30 GIOCHI PER IL CASIO

Sandro Del Bello e Anna Paganini

GRUPPO EDITORIALE JACKSON

30 GIOCHI PER IL CASIO

Sandro Del Bello
Anna Paganini

GRUPPO
EDITORIALE
JACKSON
Via Rosellini, 12
20124 Milano

SOMMARIO

INTRODUZIONE

Dal titolo avrai già capito che il libro che ti accingi a leggere è un tentativo di conciliare gioco e apprendimento. Secondo noi è possibile fare pratica nella programmazione anche attraverso divertenti e semplici passatempi che rendono senz'altro la materia più interessante e piacevole.

Potrai infatti alternare momenti di "studio" ad altri in cui avrai un po' di relax utilizzando il tuo CASIO come strumento di gioco.

Il libro non è un manuale di avviamento al BASIC, ma un naturale proseguimento e complemento di quello fornito in dotazione al momento dell'acquisto del computer. È quindi rivolto soprattutto a coloro che già possiedono le nozioni di base; questo naturalmente non impedirà il suo uso ai lettori che le hanno dimenticate o ne sono sprovvisti. L'appendice A al termine del libro consentirà infatti anche a questi ultimi una proficua lettura delle spiegazioni contenute nel volume.

I giochi presentati sono di vario tipo e mettono alla prova, a seconda dei casi, i tuoi riflessi, la tua memoria, le tue capacità logiche ed anche la tua buona stella.

Qualcuno lo conoscerai senz'altro, mentre altri sono originali; nel loro insieme consentiranno di cimentarti in sfide con te stesso, il computer o il tuo amico preferito.

Ogni capitolo comprende:
— breve presentazione del gioco
— modalità di utilizzo
— listato
— spiegazioni relative all'algoritmo impiegato con eventuali note al listing

Per quanto possibile si è cercato di ordinare i programmi tenendo conto del relativo grado di difficoltà o quanto meno attuando una logica progressione degli argomenti.

Completano la raccolta tre appendici che riguardano nell'ordine il Basic del PB-100, qualche semplice particolarità ed infine alcune considerazioni sui diversi messaggi di errore.

Sperando nell'utilità di queste pagine ti auguriamo buon lavoro ma soprattutto buon divertimento.

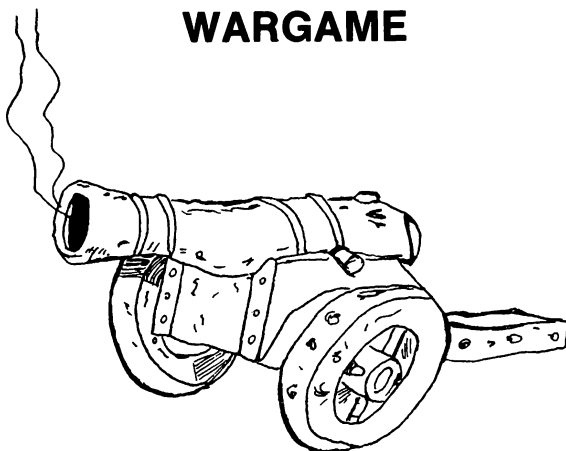
Gli Autori

AVVERTENZE

— In tutti i programmi il simbolo "ƀ" (blank) è stato utilizzato per indicare la presenza di uno spazio vuoto (SPC).

— Se qualche riga sembra troppo lunga e impossibile da digitare a fianco di uno stesso numero di linea, non si tratta di un errore. Per poterla scrivere al completo basterà compattarla, eliminando cioé gli spazi che il computer provvede automaticamente ad inserire tra le parole chiave e i vari caratteri. A tale scopo occorre posizionare il cursore sopra gli spazi in eccesso e premere quindi il tasto DEL.

— Al termine di ogni listato, per una tua maggiore comodità, è riportato il numero di passi che restano inutilizzati (nella versione base). Potrai così controllare se il programma è stato battuto correttamente o meno.

WARGAME

Per iniziare eccoti un gioco che, dopo qualche prova per familiarizzare con il suo meccanismo, non mancherà di coinvolgerti in serrate disfide contro i tuoi compagni delle ore libere. Il tuo Paese è in guerra con il nemico invasore e dalle linee avanzate della trincea il comandante ti ha forzatamente inviato ad una missione che potrebbe decidere le sorti della battaglia. Da una postazione ottimamente difesa il cannoniere avversario ha costretto la tua compagnia sulle difensive; l'unico modo per liberarsi dall'incomoda situazione è quello di raggiungere un altro cannone posto nelle vicinanze e cercare di abbattere il nemico.

Coraggio, aggiusta in fretta la mira se vuoi rivedere la tua bella!

COME SI GIOCA

È un gioco per due persone che devono a turno lanciare un proiettile col cannone, cercando di colpirsi vicendevolmente. Il giocatore di sinistra (che è poi il primo ad effettuare il tiro) è rappresentato dal carattere Ω; quello di destra da ¥. Scegli, in accordo con il tuo amico-avversario, uno dei due

simboli che ti impersonerà per tutta la gara.

Dopo la presentazione dei contendenti (PLAYER 1 Ω PLAYER 2 ¥) sul pannello comparirà l'indicazione che la gara è aperta per il giocatore di turno.

Questi, premendo EXE, otterrà la visualizzazione dei dati la cui conoscenza si rende necessaria per un tiro correttamente impostato e maggiore probabilità di successo. Fai quindi particolare attenzione a ciò che ti viene mostrato poiché messo in rapporto con l'esito della prova ti fornirà utili indicazioni per i successivi tentativi.

Il moto del proiettile è infatti influenzato dal vento attualmente presente, risentendo fortemente sia della sua intensità che della direzione verso cui spira.

Questi dati verranno stampati ad esempio sotto la forma

WIND= 2 m/s→

che indica una velocità di 2 metri al secondo verso destra (vento a favore del primo giocatore).

Si dovranno a questo punto inserire la velocità che si intende dare al proiettile (espressa in m/s) e l'angolo di tiro, compreso fra 0° e 90°. Entrambi questi valori possono essere espressi anche con numeri decimali ed anzi ciò risulterà spesso necessario per fare centro.

Il proiettile partirà, dopo aver premuto EXE, seguendo l'effettiva traiettoria risultante dai dati inseriti e dalle già riportate condizioni del vento.

Sul display questa curva parabolica sarà rappresentata con cifre che, partendo dal giocatore che ha effettuato il lancio, compariranno in successione verso il nemico. Questi numeri si riferiscono alla distanza dell'ordigno esplosivo del suolo; distanza che varia istante per istante.

Per vincere, la parabola di tiro dovrà terminare esattamente nello spazio occupato dall'avversario; quindi le cifre rappresentanti la traiettoria dovranno crescere per un certo tratto e poi decrescere fino a raggiungere quota 0 dal suolo proprio in corrispondenza della postazione nemica.

Non credere che sia troppo facile eseguire il tiro esatto che permette la vittoria; dovrai senz'altro fare diversi tentativi prima di riuscire a mettere correttamente in relazione la velocità impressa al proiettile e l'angolo di lancio con le condizioni del vento. Il più delle volte il tiro sarà inesatto, corto o lungo, nei casi in cui la parabola termini prima o teoricamente oltre il simbolo opposto (situazione che non può ovviamente venire visualizzata sul display).

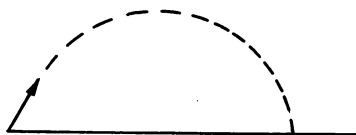
Ecco alcuni esempi:

Una buona memoria nel ricordare i tentativi già effettuati ti sarà utilissima per battere sul tempo l'altro giocatore.

Ovviamente le regole impongono di non servirti di carta e penna né per riportare gli esiti e i valori delle varie prove, né per operare calcoli di alcun tipo.

Giocando potrai notare che la velocità del vento rimane costante per due tiri consecutivi, uno per ogni giocatore; naturalmente a seconda della direzione della freccia i dati dovranno essere interpretati in modo diverso (vento favorevole o contrario) dai due avversari

WIND= 3 m/s → vento favorevole al giocatore di sinistra e contrario a quello di destra

Affinchè tu possa avere subito una certa disinvoltura, ti consigliamo di non inserire velocità troppo elevate; non scendiamo maggiormente in dettagli per non privarti del gusto della scoperta. Per quanto riguarda l'angolo di tiro, come potrai verificare, l'angolo di 45° determina la massima gittata.

Ultima notazione prima del via riguarda ancora il valore relativo al vento; questo risulta sempre un numero intero compreso fra 0 e 5. In realtà ciò corrisponde ad un arrotondamento ed è per questo motivo che, nelle stesse condizioni, un tiro precedentemente diretto sul bersaglio, può invece risultare errato; ciò eviterà un gioco troppo scontato e ripetitivo.

IL LISTATO

```
1    $="0123456789": G$="PLAYER": B$="Ω": C$=" ¥"
2    PRINT "*ƀWAR-GAME ƀ*": FOR I=1 TO 2: GOSUB 50: GOSUB 90:
     NEXT I
3    A=RAN#*11-5: FOR I=1 TO 2: GOSUB 60
4    PRINT CSR 2; G$; I: GOSUB 90
5    PRINT: PRINT "WIND="; ABS(INT A); "m/sƀ→";: IF A<0;
     PRINT CSR 11; "←";
6    GOSUB 90: PRINT: INPUT "VELOCITA'",V: V=ABS V
7    INPUT "ANGOLO",F: IF F<0; F=0
8    IF F>89; F=89
9    GOSUB 60: X=1: IF I=2; A=-A
10   T=X/(V*COS(F)+A)
11   Y=INT(V*SIN(F)*T-9.8*T*T/2+.5): IF Y>9; Y=9
12   IF Y<0; Y=0
13   Z=X: IF I=2; Z=11-X
14   PRINT CSR Z; MID(Y+1,1);: IF X=11 THEN 17
15   IF Y=0; PRINT: PRINT "TIRO CORTO!!";: GOSUB 90: NEXT I:
     GOTO 3
16   X=X+1: GOTO 10
17   IF Y≠0; PRINT: PRINT "TIRO LUNGO!!";: GOSUB 90: NEXT I:
     GOTO 3
18   PRINT CSR Z; "■";: GOSUB 90
19   PRINT: PRINT "THE WINNER ƀ:";: GOSUB 90: GOSUB 50: STOP:
     GOTO 3
50   PRINT: PRINT G$; I;"ƀƀ";A$(I);: RETURN
60   PRINT: PRINT B$; CSR 11; C$;: RETURN
90   FOR J=1 TO 300: NEXT J: RETURN
```

Il segno Ω nella linea 1 è fatto con MODE. SHIFT N
Il segno ¥ nella linea 1 è fatto con MODE. SHIFT R
Il segno → nella linea 5 è fatto con MODE. SHIFT P
Il segno ← nella linea 5 è fatto con MODE. SHIFT I
Il segno ■ nella linea 18 è fatto con MODE. SHIFT Z

NOTA: nel presente listato e in tutti quelli che seguiranno il segno ƀ indica la battitura di uno spazio (SPC).
(restano 2 passi).

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A	velocità del vento
F	angolo di tiro
T	calcolo intermedio
V	velocità assegnata al proiettile
X	numero progressivo delle stampe
Y	quota del proiettile
Z	posizione di stampa

I primi giochi che ti presentiamo hanno un algoritmo piuttosto lineare e questo ci ha consentito di soffermarci maggiormente sul modo talvolta un po' particolare di utilizzare determinate istruzioni. Iniziamo subito dalla linea 1.

L'uso della variabile $ per immagazzinare una serie di caratteri (in questo caso le 10 cifre) per poi estrarli successivamente tramite la funzione MID, è un accorgimento che spesso ricorrerà nei vari listati. Qui si doveva ottenere la rappresentazione della traiettoria del proiettile con la stampa delle diverse quote dal suolo in tutti gli spazi del display. Saprai certamente che la visualizzazione di una cifra con le variabili numeriche richiede tuttavia due posizioni, una delle quali relativa al segno (spazio vuoto in caso di valore positivo). Sul display le quote sarebbero così apparse:

Ω		1		2		3		3		2	¥

Evidentemente non era il risultato voluto e una possibile alternativa è stata quella di considerare le varie cifre come semplici caratteri facenti parte della variabile $.

$="0123456789"

Eseguito il calcolo (linea 11) che fornisce l'altezza momentanea del proiettile, è stato sufficiente stampare non la quota stessa (Y), ma bensì il carattere corrispondente contenuto in $. Naturalmente il valore massimo di Y è stato limitato a 9 (IF Y>9; Y=9). L'estrazione voluta si effettua con la funzione MID che seleziona in questo caso il carattere che occupa in $ la posizione Y+1 (linea 14). Così facendo tutti gli spazi del pannello risulteranno occupati dalle cifre.

Proseguendo nella lettura del listato si incontra nella linea 2, relativa alla presentazione dei giocatori, un rimando alla subroutine 90. Questa, più volte utilizzata nel corso del programma, consiste semplicemente in un cosiddetto "ciclo vuoto", cioè un ciclo che non comprende al suo interno alcuna istruzione. Qual è allora il suo significato?

Il computer, per contare qui da 1 a 300 impiega un certo tempo prima di ritornare al programma principale. Questo ritardo è necessario al fine di mantenere visualizzata nel display una scritta prima della stampa successiva in modo che l'utilizzatore riesca a leggerla comodamente.

Display	
PLAYER 1 Ω	1ª visualizzazione
PLAYER 1 Ω	la scritta rimane stampata il tempo necessario al computer per contare da 1 a 300
PLAYER 2 ¥	2ª visualizzazione

Una soddisfacente perdita di tempo (né troppo breve né eccessivamente elevata) risulta necessaria ad una buona realizzazione di tutti i programmi, soprattutto dei giochi, dove l'aspetto grafico riveste sempre una particolare importanza. Nell'esempio riportato, tra la presentazione di un concorrente e quella dell'altro, non è stato necessario cancellare il display, in quanto le due scritte si sovrappongono ed occupano esattamente gli stessi spazi.

Se così non fosse occorrerebbe far precedere alla seconda stampa la "pulizia" del pannello. Questa operazione si effettua tramite l'uso della sola parola PRINT non seguita da alcun dato. Si ha quindi, ad esempio, la successione (linee 4/5)

Display	
Ω PLAYER 1 ¥	1ªstampa
Ω PLAYER 1 ¥	perdita di tempo
	cancellazione (PRINT)
WIND= 2m/s →	2ª stampa

Nella linea 3 viene calcolata casualmente l'intensità del vento facendo uso della funzione RAN # che certamente ti è già nota. La sua moltiplicazione per 11 consente di ottenere un numero compreso tra 0 e 11 (estremi esclusi) e la successiva sottrazione di 5 fornisce una velocità del vento che va da -4,999... a 5,999....

Non ci soffermeremo sulla funzione ABS che qui non presenta particolarità, né tanto meno (linee 10,11) sulla formula relativa al calcolo della traiettoria del proiettile; prendila per buona o, se vuoi approfondire la cosa, ricerca nel tuo libro di fisica il paragrafo sull'argomento.

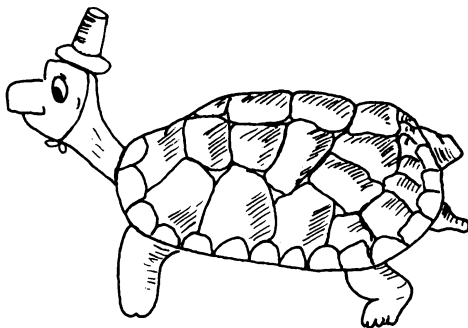
Tutto bene fino qui?

Mancano ancora un paio di chiarimenti alla conclusione di questa parte. Il primo riguarda le altezze del proiettile la cui stampa, come avrai visto, inizia alternativamente da uno o dall'altro lato del display. In entrambi i casi la visualizzazione delle cifre avviene tramite un'unica PRINT, utilizzando la variabile Z, indicante la posizione di stampa (linea 14).

Al turno del primo giocatore Z assume il valore della variabile X, mentre per il secondo Z è uguale a 11-X. Poichè X rappresenta il numero progressivo delle stampe, ciò consente di avere in un caso un incremento di Z (Z=1..2..3..) e nell'altro un decremento (10..9..8..). Così facendo la traiettoria prenderà il via dallo spazio 1 per Ω e dalla posizione 10 per ¥.

Un'ultima notazione riguarda la parola chiave STOP, nella linea 19. Questa istruzione non certo frequente consente di fermare l'elaborazione del programma che verrà ripresa soltanto dopo aver premuto il tasto EXE.

Il suo utilizzo si è reso conveniente dal momento che la stampa contenuta nella subroutine 50 terminava con ";" e il via ad un'altra prova (GOTO 3), omettendo il comando di arresto, avrebbe avuto luogo senza alcuna pausa.

SPRINT

Non poteva certo mancare nella nostra raccolta un gioco che mettesse alla prova in modo specifico i riflessi.

Vuoi sapere se sei un tipo "sveglio" oppure se annoveri tra i tuoi parenti lumaconi e tartarughe?

Al CASIO.... l'ardua sentenza.

....COME SI GIOCA

Il gioco consiste dunque in tre test, al termine dei quali potrai avere una valutazione dei tuoi riflessi. Ti avvertiamo che siamo stati un po' "cattivi" e se vuoi ottenere risultati discreti dovrai impiegare la massima prontezza. La risposta agli stimoli visivi che ti sono proposti dovrà essere istantanea ed immediata; solo così eviterai di vedere comparire nel display valutazioni negative, del tipo "SCARSO" o "PESSIMO". Anche in questo caso però non ti scoraggiare e riprova, ponendoti come traguardo l'ambito e quasi irraggiungibile "OTTIMO".

Presta ora molta attenzione a ciò che ti viene richiesto nelle singole prove,

ognuna delle quali sarà preceduta dalla relativa scritta ed avrà inizio in seguito alla pressione di "EXE".

Nella prima gara il computer sceglie a caso una lettera fra le ventuno che compongono l'alfabeto italiano e la fa comparire per un istante al centro dèl pannello di lettura. Il gioco consiste nel premere il più velocemente possibile il tasto corrispondente alla lettera visualizzata.

Questa prova risulta molto impegnativa in quanto non è facile compiere l'operazione descritta nel tempo consentito, che è davvero brevissimo. È necessaria una formidabile prontezza di riflessi ed anche la conoscenza della posizione delle varie lettere ti sarebbe di grande aiuto; ti consigliamo in ogni caso di partire con le mani già vicine alla tastiera.

Se non riesci a superare la prova ti sono assegnati 10 punti di penalità, penalità che diminuiscono quanto più velocemente hai premuto il tasto esatto. Per quanto ci riguarda tieni conto che non è stato mai possibile scendere al di sotto dei cinque, sei punti. E a te come è andata?

I tuoi riflessi sono stati rapidissimi o troppo lenti? Stai tranquillo, sei ammesso comunque alla seconda prova, che forse è un po' più semplice.

Vedrai susseguirsi rapidamente le cifre dallo 0 al 9 e la tua abilità ti dovrà consentire di premere il tasto corrispondente al numero 9 esattamente quando tale cifra compare sul display. Fai attenzione: la pressione del tasto deve essere piuttosto decisa, altrimenti i numeri continueranno a ruotare.

Dovrai cimentarti in tre manche e lo scorrere delle cifre sarà sempre più rapido. Ogni errore ti costerà cinque penalità, per cui se sei proprio lento come una tartaruga sarai penalizzato di ben 15 punti che andranno a sommarsi a quelli realizzati precedentemente.

Ti possiamo dare un consiglio: mentre la prima rotazione è piuttosto lenta e puoi quindi schiacciare il tasto proprio quando vedi comparire il numero 9, nelle altre due, data la maggior velocità, ti conviene premerlo un istante prima che si visualizzi.

Eccoti ora alla Prova n. 3, nella quale un quadrato bianco "□" salta da uno spazio all'altro del display, rimanendo fermo solo per un attimo.

Ed è proprio in quell'istante che dovrai fermarne la corsa, premendo il numero da 0 a 9 corrispondente allo spazio in cui si trova. Questa volta non vi è un massimo di penalità, in quanto il "□" continuerà a muoversi fino a quando non sarai riuscito a colpirlo; ogni salto ti costa però ben 5 punti.

Perciò in quest'ultima prova dovrai mettere ancora più concentrazione, altrimenti ti potresti facilmente ritrovare con un numero altissimo di penalizzazioni ed otterresti quindi una valutazione negativa, anche se nelle prove precedenti avessi dimostrato ottimi riflessi.

Il livello della tua prontezza apparirà al termine premendo EXE, secondo lo schema seguente:

da	0	a	7	penalità	→	OTTIMO
da	8	a	15	penalità	→	BUONO
da	16	a	23	penalità	→	NORMALE
da	24	a	31	penalità	→	SCARSO
da	32	in	poi		→	PESSIMO

IL LISTATO

FUORI PROGRAMMA IN MODO RUN:
R$(0) = "OTTIMO"
R$(1) = "BUONO"
R$(2) = "NORMALE"
R$(3) = "SCARSO"
R$(4) = "PESSIMO"

```
 1   PRINT "RIFLESSI";: L=200: GOSUB 30
 2   Q=0:D=0: $="ABCDEFGHILMNOPQRSTUVZ": H=
     INT(RAN#*21)+1
 3   A$=MID(H,1): $=" ъ Penalità": N$="TOTALE=":
     PRINT "Prova n° 1"
 4   GOSUB 30: PRINT CSR 6; A$;
 5    FOR O=1 TO 39: IF KEY≠A$; NEXT O: PRINT CSR 0;
     "Troppo tardi";
 6   P=INT(O/4)
 7   PRINT P; $;: GOSUB 30: PRINT "Prova n° 2": FOR F=
     15 TO 5 STEP -5
 9   FOR C=0 TO 9: L=F: GOSUB 30: PRINT CSR 6; C;
10   IF KEY ≠"9"; NEXT C: GOTO 9
11   L=150: IF C=9; GOSUB 30: PRINT CSR 0; "Bene! ъ0";
     $;: GOTO 13
12   GOSUB 30: PRINT CSR 0; "No! ъ5"; $;: Q=Q+5
13   L=200: GOSUB 30: NEXT F: PRINT N$; Q;: GOSUB 30
14   PRINT "Prova n° 3": GOSUB 30
15   A=INT(RAN#*10): PRINT CSR A; "□";
16   FOR I=1 TO 25: B$=KEY: IF B$≠""; IF VAL(B$)=A THEN 18
17   NEXT I: PRINT: D=D+5: GOTO 15
18   D=D+INT(I/5): W=P+Q+D: PRINT CSR 0; "ъъ Colpito! ъ";
     D; $, N$; W;
19   GOSUB 30: IF W>30; W=39
20   PRINT R$(INT(W/8)): GOTO 2
30   PRINT: FOR O=1 TO L: NEXT O: RETURN
```

Nella riga 15 il segno □ è fatto con MODE. SHIFT S
In tutto il listato alcune scritte sono in minuscolo (MODE.)
(restano 0 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A	spazio di stampa del "□"
A$	lettera estratta
D	penalità della III prova
P	penalità della I prova
Q	penalità della II prova
R$(0)...R$(4)	valutazioni dei tuoi riflessi
W	penalità totali

La prima particolarità che avrai forse notato nel leggere il programma è l'inserimento delle valutazioni relative ai riflessi in modo RUN.

Il listato occupa infatti tutta la memoria disponibile, il che ha impedito di porle insieme alle altre istruzioni, dopo un regolare numero di linea. Questa tecnica, adottata anche in altri giochi, consente ugualmente di definire e memorizzare alcuni dati e di poter quindi richiamarli durante lo svolgimento del programma. L'assegnazione delle valutazioni in oggetto alle variabili stringa R$(0)....R$(4) al posto delle corrispondenti R$....V$ è stata effettuata non certo per crearti inutili complicazioni, ma soltanto per rimarcare la loro appartenenza ad uno stesso vettore.

Un altro accorgimento usato per risparmiare non molti ma essenziali passi di programma è stato quello di memorizzare alcune scritte che ricorrono più di una volta in variabili di carattere che all'occorrenza verranno richiamate.

Ad esempio la stringa "PENALITA' " è contenuta in $, non appena quest'ultima cessa di indicare le lettere dell'alfabeto (linea 3).

Riteniamo che la comprensione della prima prova non presenti particolari difficoltà, specialmente dopo i chiarimenti, dati in precedenza, riguardo all'estrazione di caratteri dalla variabile $.

Leggermente più difficoltosa è la Prova n. 2, dove si doveva variare la velocità di rotazione delle cifre.

L'algoritmo consta di due cicli, uno interno all'altro: quello esterno presenta un incremento negativo di 5, ed F assume quindi i valori 15, 10, 5 che rappresentano il tempo durante il quale ogni cifra rimane visualizzata sul display. Così ogni manche risulterà leggermente più veloce della precedente.

Il ciclo interno su C determina il susseguirsi dei numeri dallo 0 al 9 ed ogni stampa è preceduta da un rimando alla subroutine 30, dove si verifica un ritardo pari al valore attuale di F (linea 9 → L=F: GOSUB 30).

Per quanto riguarda la terza ed ultima prova, troverai nella linea 16 due IF consecutivi: è chiaro che l'uscita dal ciclo avverrà soltanto nel caso che le due condizioni si verifichino entrambe; quando cioè premerai un tasto (1ª condizione) ed esso indicherà correttamente la locazione del quadrato "□" da colpire (2ª condizione).

Ci sembra infine utile soffermarci su come nelle tre prove vengono calcolate le penalità da assegnare.

Nella linea 6 il tempo impiegato a cercare e battere la lettera stampata, indicato dalla variabile O, viene diviso per 4. Poichè al termine del ciclo O assume il valore 40, male che ti vada sarai penalizzato di 10 punti.

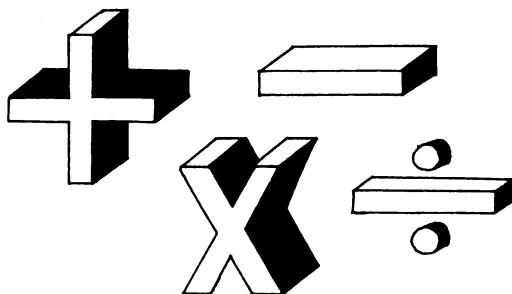
Nella prova centrale ogni errore ti costerà un incremento di 5 nel punteggio (Q=Q+5 nella linea 12).

Nell'ultimo gioco non soltanto ti "prenderai" 5 penalità tutte le volte che il quadrato "□" salta da uno spazio all'altro, ma avrai anche una piccola aggiunta proporzionale al tempo (I) della sua permanenza in un determinato posto, prima di essere bloccato. Essendo il massimo valore di I uguale a 25, esso viene in seguito diviso per 5 per non darti eventualmente penalità superiori a quelle stabilite in caso di insuccesso.

Potrai a questo punto vedere se sei stato veramente abile come immodestamente ti ritieni o se, al contrario, hai bisogno di un buon bagno di umiltà.

Infatti nelle linee 19 e 20, in relazione al punteggio accumulato, vengono valutati i tuoi riflessi.

Le penalità totalizzate in tutte le prove, limitate ad un massimo di 39, sono divise per 8; ciò consente di ottenere un numero che varia da 0 a 4 che indicherà l'elemento del vettore R$ da visualizzare.

QUICK

Non voltare subito pagina, anche se questo gioco ti potrebbe far pensare all'ultimo compito in classe di matematica e alle numerose operazioni ancora da eseguire, mentre l'ora concessa dal professore sta inesorabilmente scadendo.

Anche qui sarai alle prese con i numeri ma, contro ogni apparenza, il divertimento è assicurato. Leggi le spiegazioni e saprai esattamente che cosa ti attende.

....COME SI GIOCA

Il gioco, che noi abbiamo chiamato "Quick", consiste in una prova di velocità nell'eseguire le quattro operazioni.

È scontato che tu sappia addizionare, sottrarre, moltiplicare e dividere, ma non è detto che ciò ti riesca in un tempo limitatissimo che non concede alcuna incertezza o esitazione.

La fretta e l'ansia causate dal veder scorrere velocissimi i secondi ti porteranno probabilmente a commettere errori che certo altrimenti non

faresti mai. Ma, d'altra parte, è solo in una situazione come questa che puoi provare a te stesso se hai davvero dimestichezza con i numeri. Sarà divertente anche giocare con gli amici e misurare insieme la vostra abilità e prontezza nel calcolo.

Il computer ti propone in successione le quattro operazioni, nell'ordine + _ x :. I numeri vengono scelti da lui casualmente e in tutte le quattro prove saranno compresi fra 300 e 900.

Ogni volta si visualizza sul display l'operazione da eseguire che, dopo qualche istante, scompare per lasciare posto, nell'estremità destra del pannello, ad un contatore del tempo che cresce velocemente.

A questo punto devi risolvere il calcolo che ti è stato proposto e quindi inserire il tuo risultato facendo attenzione a premere, subito dopo l'ultima cifra, il tasto corrispondente al segno "=". Ciò bloccherà il contatore che ha continuato nel frattempo ad avanzare.

Devi sapere che per l'addizione e la sottrazione il tempo massimo consentito è di 100, mentre per moltiplicazione e divisione è stato aumentato fino a 300. Nel caso tu, inavvertitamente, commetta qualche errore di battitura, se ancora non hai bloccato il tutto tramite "=" ti basta premere il tasto " E ": ciò che hai scritto verrà cancellato e potrai così inserire nuovamente il numero che ritieni esatto. La cosa comporta però una notevole perdita di tempo, per cui ti conviene prestare molta attenzione nella scrittura iniziale.

Il computer ti informa quindi dell'esattezza o meno del calcolo da te eseguito, se ovviamente i secondi a tua disposizione non sono ancora terminati; in quest'ultimo caso ti avverte con un "TROPPO TARDI". Ora premi EXE e passa all'operazione successiva. Al termine del gioco appariranno sul display gli errori da te commessi ed il tempo impiegato complessivamente nelle quattro prove.

Ogni risultato errato ed ogni "TROPPO TARDI" ti costeranno ben 400 penalità; se invece il risultato è esatto ti verrà assegnato un punteggio uguale al tempo utilizzato a risolvere l'operazione, cioè al numero su cui hai bloccato il contatore.

Proseguiamo ora con un paio di osservazioni e consigli.

Se nella sottrazione il primo termine risulta minore dell'altro, il risultato avrà ovviamente segno negativo che non devi dimenticare di inserire. Nella divisione il quoziente è stato approssimato a meno di 0,01 per cui sarà sufficiente scrivere solo le prime due cifre decimali.

È forse impossibile risolvere a mente le ultime due operazioni (x, :); dovrai così ricopiare velocemente il testo ed eseguirle con l'aiuto di carta e penna. Potresti invece risparmiare del tempo prezioso nell'addizione e nella sottrazione, se riuscissi a calcolarne mentalmente il risultato nei brevissimi istanti

in cui rimangono visualizzate sul pannello di lettura. Così, appena il contatore inizia a girare, avresti già pronto il numero da inserire.

Non sempre ti sarà possibile ma vale la pena di tentare, soprattutto se la fortuna ti aiuta e il computer sceglie per te cifre piuttosto semplici.

IL LISTATO

```
  1   VAC
  2   $="OPERAZIONE ƀ": M$=" + ": N$="-": O$="x": P$=":":
      PRINT "QUICK"
  3   PRINT $; M$;: GOSUB 30: D=B+C: Y=100: GOSUB 50: PRINT
      $; N$;: GOSUB 30
  5   D=B-C: GOSUB 50: PRINT $; O$;: GOSUB 30: D=B*C: Y=300:
      GOSUB 50
  6   PRINT $; P$;: GOSUB 30: D=INT(B/C*100)/100: GOSUB 50
  7   PRINT: PRINT "ERRORI:"; 4-X;: FOR I=1 TO 200: NEXT I: PRINT
  8   PRINT "TEMPO:"; W: END
 30   FOR I=1 TO 2: A(I)=INT(RAN#*600)+300: NEXT I
 35   K=K+1: FOR I=1 TO 400: NEXT I: PRINT: PRINT B; L$(K); C;:
      RETURN
 50   FOR I=1 TO 400: NEXT I: PRINT: F$="": Z=0
 60   FOR I=0 TO Y: PRINT CSR 8; I;
 61   E$=KEY: IF E$=" = "; PRINT: GOTO 90
 62   IF E$="E"; PRINT: PRINT CSR 8; I;: Z=0: F$="": GOTO 80
 63   IF E$="" THEN 80
 64   IF Z<6; Z=Z+1: PRINT CSR Z; E$;: F$=F$+E$
 65   IF KEY=E$; I=I+1: GOTO 65
 80   NEXT I: PRINT: PRINT "TROPPO TARDI": W=W+400: RETURN
 90   G=VAL(F$): IF G=D; W=W+I: PRINT "BENE! T=";
      I: X=X+1: RETURN
100   PRINT " ƀERRORE!!": W=W+400: RETURN
```

Il segno x nella riga 2 è fatto con MODE . SHIFT F

(restano 0 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
B,C	i due termini delle operazioni
D	risultato corretto del calcolo
F$	risultato da te inserito
M$,N$, O$,P$	contengono i segni delle quattro operazioni
W	punteggio totale
X	numero delle operazioni eseguite correttamente ed in tempo utile
Y	tempo a disposizione ad ogni prova
Z	spazio di stampa di ogni cifra digitata

Questo programma, pur non essendo particolarmente difficile, presenta tuttavia argomenti abbastanza delicati rappresentati da frequenti richiami alla subroutine e da un uso un po' sofisticato della funzione KEY.

Per ognuna delle quattro operazioni che devi svolgere è necessario che il computer:

(1) scelga casualmente i due numeri relativi al calcolo
(2) li visualizzi sul display unitamente al segno dell'operazione
(3) definisca il risultato
(4) azioni il contatore del tempo
(5) permetta l'inserimento del tuo risultato
(6) ne controlli la correttezza assegnando il relativo punteggio
(7) risponda adeguatamente al tuo tentativo

Come puoi notare, quasi tutti i passi della sequenza sono indipendenti dal tipo di operazione e abbiamo potuto scriverli una sola volta riunendoli in due subroutine richiamate ad ogni prova, facendo eccezione soltanto per la parte che riguarda il calcolo del risultato (punto 3).

È infatti ovvio che la tecnica varia proprio in relazione all'operazione da effettuare, determinando una corrispondente diversità, ad esempio, fra l'istruzione per l'addizione e quella per la divisione.

Potresti obiettare che in realtà anche i segni differiscono, ma la loro stampa all'interno della subroutine è stata possibile utilizzando un vettore. Infatti i simboli + - x : nella linea 2 sono stati assegnati alle variabili da M$ a P$ e successivamente (linea 35) richiamati uno alla volta, a seconda della prova,

tramite L$(K). Ad ogni operazione K viene incrementato di 1 e si ha quindi

L$(1) = M$ = "+"
L$(2) = N$ = "-"
L$(3) = O$ = "x"
L$(4) = P$ = ":"

Riassumendo, la subroutine 30 consente la scelta dei termini delle operazioni e la loro visualizzazione mentre nella subroutine 50 troviamo i punti 4) 5) 6) e 7) precedentemente descritti.

Il contatore del tempo è azionato nella linea 60 effettuando un ciclo su I da 0 ad Y. Tieni presente che il valore di Y è stabilito nel programma principale prima del rimando alla subroutine ed è uguale a 100 per l'addizione e la sottrazione, a 300 per le rimanenti operazioni. Questa necessaria differenza ci ha obbligato ad usare appunto tale variabile anzichè un valore costante.

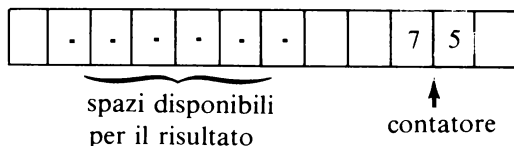
Lo scorrere del tempo viene visualizzato stampando in successione in CSR 8 i valori di I, in modo da avere costantemente la situazione sotto controllo.

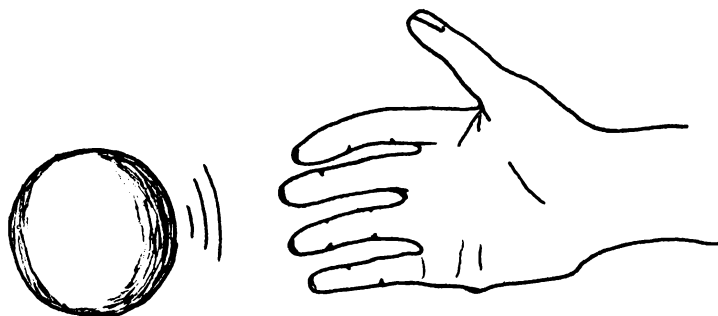
In questo caso non si è resa necessaria un'ulteriore iterazione di ritardo tra due stampe di I, poichè le istruzioni inserite all'interno del ciclo sono sufficienti a dare una velocità ragionevole.

Passiamo ora ad analizzare l'argomento relativo alla funzione KEY, che consente la memorizzazione di un carattere premuto sulla tastiera senza causare l'arresto del programma, a differenza dell'istruzione INPUT. Nel nostro caso era necessario utilizzarla per inserire il risultato del calcolo da te effettuato, consentendo contemporaneamente il funzionamento del contatore.

I problemi da affrontare erano sostanzialmente due: memorizzare e visualizzare non la singola cifra ma, ovviamente, tutto il numero da te impostato ed evitare che una prolungata pressione di un tasto determinasse la ripetuta stampa della cifra corrispondente.

La prima difficoltà è stata risolta assegnando ad E$ le singole cifre inserite (E$ = KEY nella linea 61) e quindi (linea 64) accumulandole in F$. La loro lettura si rende possibile incrementando di volta in volta la variabile Z, indicante lo spazio del display.

Il numero massimo delle cifre del risultato non è mai maggiore di 6 e Z è stata conseguentemente limitata per evitare, in caso di errore, l'interferenza con il cronometro.

Nella linea 65 troviamo invece la soluzione al secondo problema. Finchè un determinato tasto non viene rilasciato il programma è obbligato alla medesima riga e soltanto I continua ad incrementarsi. In tal modo, anche se apparentemente il contatore risulta fermo, al momento in cui cessa la pressione esso riassumerà il suo valore corretto.

ROLLING

Hai mai gareggiato al Luna Park in quel simpatico gioco dove si tira la pallina nelle buche colorate ed ogni centro fa avanzare un poco il tuo cavallo?

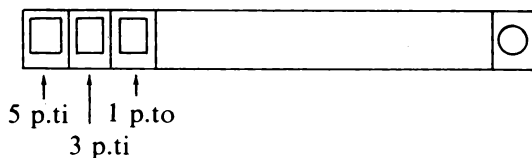
Bene! Ora prova questo "Rolling" e vedrai che è altrettanto divertente. Certo centrare le buche non è facile e dovrai effettuare tiri ben misurati, altrimenti la tua pallina rotolerà inutilmente nella pista.

....COME SI GIOCA

Come avrai già capito, devi cercare di mandare la pallina in una delle tre buche visualizzate a sinistra nel display e rappresentate da □ □ □

Per effettuare il lancio, alla comparsa della scritta "TIRA!", devi premere un tasto qualsiasi e ciò farà partire la palla.

La lunghezza del tiro è determinata dal tempo per il quale si tiene abbassato il tasto stesso ed in minima parte dal caso.
Se il lancio è corto la pallina non arriva alle buche; al contrario se è lungo passerà sulle stesse senza cadervi dentro e tornerà indietro verso il punto di partenza.

Quando una delle buche viene centrata, si annerisce e compare il relativo punteggio: 1,3 o 5 punti.

Soprattutto le prime volte faticherai non poco a individuare qual è l'esatto tempo di pressione che ti permette di effettuare un tiro dalla forza adeguata. Anche quando dopo i vari tentativi riesci nell'intento, ti sarà ugualmente difficile ripeterti poichè la giusta pressione è piuttosto breve e quindi differenze seppur minime in più o in meno possono modificare completamente il lancio. Sarai comunque tu stesso a renderti subito conto della buona riuscita o meno della prova, ancor prima della conclusione del tiro.

Infatti dalla velocità iniziale impressa alla palla potrai intuire se si tratta di un tiro errato o se hai buone probabilità di raggranellare qualche punto.

Tieni infine presente che occorre agire con prontezza perchè il gioco termina dopo un certo tempo stabilito.

Capirai bene quindi che i lanci lunghi sono più sfavorevoli e possibilmente da evitare, comportando una ovvia perdita di tempo dovuta al prolungato percorso di ritorno della palla verso la zona di lancio.

Alla fine, premendo EXE, compariranno sul pannello di lettura il numero delle buche centrate ed il punteggio totale.

IL LISTATO

```
 4   PRINT " ƀƀROLLING"
 5   VAC
10   V=100: C=.4: D=4000
30   A=20: L=L+1: GOSUB 500: PRINT CSR 4; "TIRA!";
40   L=L+1: IF KEY="" THEN 40
50   L=L+1: A=A+C*A: IF KEY≠"" THEN 50
60   L=L+1: A=INT(A/V+RAN#*3-1): IF A>22; A=22
70   L=L+1: FOR J=1 TO A
80   GOSUB 500
90   L=L+1: NEXT J: Y=ABS(A-11)
```

```
100   L=L+1: IF Y⩾3 THEN 120
110   B=B+1: P=5-2*Y: T=T+P: PRINT CSR Y; "■"; CSR 4;
      "PUNTI="; P;
115   FOR K=1 TO 200: NEXT K
120   L=L+1: J=0: IF L<D THEN 30
130   PRINT: PRINT CSR 4; "FINE"
140   PRINT CSR 0; "BUCHE"; B; "ƀƀƀƀ PUNTI"; T: GOTO 5
500   FOR K=0 TO (22+J-A)↑4/9999: L=L+1: NEXT K
510   Z=11-J: IF J>11; Z=J-11
520   PRINT: PRINT CSR 0; "□□□"; CSR Z; "○";: RETURN
```

Il segno ■ nella riga 110 è fatto con MODE . SHIFT Z
Il segno □ nella riga 520 è fatto con MODE . SHIFT S
Il segno ○ nella riga 520 è fatto con MODE . SHIFT A

(restano 133 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A	numero degli spazi che la palla deve percorrere
B	numero di buche centrate
D	tempo a disposizione
L	tempo trascorso
P	punteggio parziale
T	punti totalizzati
Z	posizione di stampa della palla

Osservando il listato ti sarai senz'altro chiesto il perchè dell'istruzione L=L+1 che ricorre diverse volte in tutto il programma. La tua curiosità è presto soddisfatta.

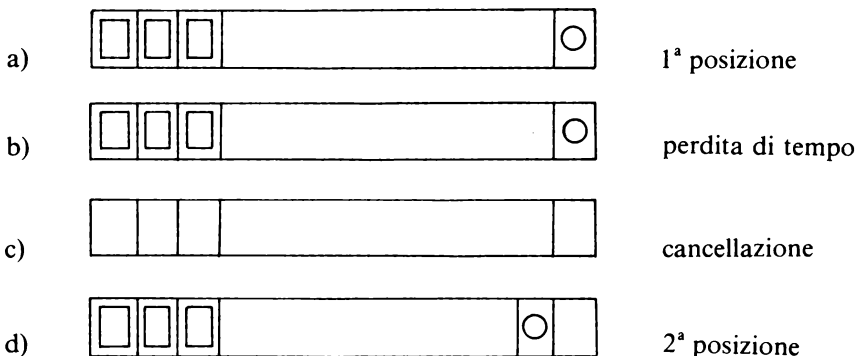
La variabile L indica il trascorrere del "tempo" durante il corso della prova, che termina quando il suo valore, a forza di successivi incrementi, si eguaglia a quello di D, posto inizialmente a 4000.

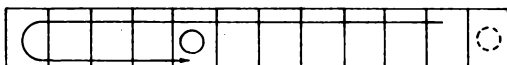
Nel programma compare per la prima volta la simulazione di un movimento che riguarda, in questo caso, la pallina.

Cercheremo quindi di spiegarti come è possibile rendere tale effetto in modo che tu possa, all'occorrenza, utilizzarlo.

Il moto della palla è stato realizzato tramite la successione di varie fasi che si riferiscono:

a) visualizzazione della palla in una determinata posizione
b) iterazione di ritardo ottenuta con un ciclo su K (linea 500)
c) cancellazione del display (PRINT)
d) successiva stampa della palla nella nuova posizione, ritorno alla fase b) e così di seguito.

In questo modo si riceve l'illusione che la palla scorra da uno spazio all'altro e sia realmente in movimento.

La pressione di un qualsiasi tasto provoca la partenza della pallina e la sua durata determina il numero degli spazi che quest'ultima dovrà percorrere. Tale valore è assegnato alla variabile A mediante i calcoli svolti nelle linee 50 e 60. L'uso del fattore moltiplicativo C (0.4) trovato sperimentalmente, fa in modo che piccole variazioni nel premere causino apprezzabili differenze nel tiro; infatti l'incremento di A avviene in modo direttamente proporzionale al suo stesso valore (A=A+C*A).

Con un tale calcolo A risulta però molto elevata ed è stata quindi corretta dividendola per V (pari a 100); inoltre le è stata assegnata una leggera percentuale di casualità allo scopo di rendere il gioco meno semplice. Difficilmente la palla compirà tutto il percorso possibile che è di 22 spazi; ovviamente tale limite non potrà mai essere superato (IF A>22; A=22)

Se A=15 la palla raggiunge il bordo, quindi torna indietro e si ferma nella posizione indicata.

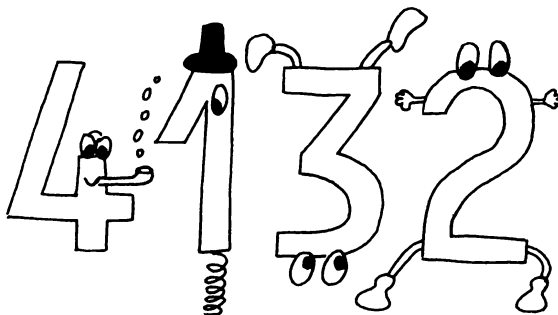
Le quattro fasi precedentemente illustrate relative al moto della pallina, sono tutte inserite nel ciclo su J (linea 70) che rimanda alla subroutine 500 e non sarà difficile individuarle in essa. L'unica linea che potrebbe crearti delle perplessità è la 500, relativa all'iterazione di ritardo (fase b). Devi sapere che infatti la maggiore difficoltà incontrata nella realizzazione di questo programma è stata senz'altro quella di rendere il più fedelmente possibile il movimento della palla, con una maggiore spinta iniziale ed una progressiva decelerazione fino all'arresto.

Questo problema è stato risolto nella linea in questione, dove viene calcolata l'attesa tra due successive posizioni di stampa. La durata di questa perdita di tempo, ottenuta tramite il ciclo su K, non è mai la stessa e ciò corrisponde ovviamente ad una diversa velocità della palla.

L'indice massimo del ciclo, (22+J-A)↑4/9999, che per comodità chiameremo M, risulta dipendente da J ed A, cioè dalla posizione della pallina e dalla potenza del tiro. La formula, seppur complicata, consente di avere un minimo ritardo quando A assume il valore massimo di 22 e la palla è appena stata lanciata (J=1); in questo caso infatti M risulta uguale a 1 E-04.

Allo stesso modo la massima perdita di tempo corrisponde all'arresto della palla, indipendentemente dalla potenza del tiro, e cioè quando il valore di J ha eguagliato quello di A; in questo caso M raggiunge il valore massimo di 22↑4/9999.

La strana elevazione a potenza è stata utilizzata per rendere meno uniforme il moto della pallina: essa infatti accentua la differenza tra i diversi valori che può assumere M.

MASTER MIND

"Una contesa di astuzia, logica ed ingegno": è la definizione che è stata data all'ormai notissimo gioco del Master Mind.

In effetti queste tre componenti della nostra "intelligenza" devono essere messe tutte in azione se vogliamo riuscire a risolvere questo gioco-rompicapo. Con un po' di esercizio, piena concentrazione ed anche, diciamolo, un pizzico di fortuna, potrai senz'altro diventare "una grande mente".

....COME SI GIOCA

Come ben saprai il Master Mind originale è una sfida tra due giocatori, ognuno dei quali deve decifrare il codice segreto dell'altro; codice che si ottiene mediante una qualsiasi combinazione di sei pioli dai colori diversi.

Quello che noi ti presentiamo è invece una versione che utilizza i numeri, non potendo ovviamente usufruire dei colori e forse è ancora più complessa poichè richiede un grado maggiore di astrazione. Non ci è stato inoltre possibile realizzare il gioco come una gara tra due persone: i passi di memoria disponibili non sono stati infatti sufficienti a far sì che il CASIO fosse contemporaneamente decifratore e codificatore.

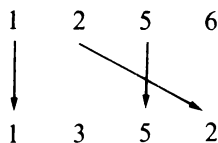
Per questa ragione, nel nostro Master Mind, al computer spetterà il compito di formare con i numeri la sequenza segreta e a te quello di indovinarla, facendo ricorso a tutte le tue capacità logiche.

Non per questo, credici, il gioco sarà meno divertente e interessante. Ma scendiamo in dettaglio.

Il calcolatore sceglie a caso quattro cifre comprese tra 1 e 6, ognuna diversa dalle altre e che, naturalmente, ti rimarranno nascoste. A questo punto ti sarà richiesto di indovinare la combinazione da lui "pensata" e tu dovrai cercare di riprodurre gli stessi numeri nelle medesime posizioni.

Premi quindi EXE e il tuo amico-avversario CASIO ti darà delle preziose informazioni visualizzando, a destra del numero da te impostato, quadrati bianchi o neri che hanno un significato ben preciso. Ogni quadrato nero sta ad indicare una cifra uguale a quella del codice segreto ed esattamente nella stessa posizione; il quadrato bianco rappresenta una cifra uguale ma situata in un altro posto. Semplice, no?

Ma, se hai ancora qualche dubbio, questa esemplificazione ti renderà tutto chiaro. Poniamo che il computer abbia selezionato 1 2 5 6 : se tu inserisci 1352 sul pannello si visualizzerà

Infatti, come vedi, i due quadrati neri indicano che l'1 e il 5 sono esatti e si trovano nella stessa collocazione, mentre il quadrato bianco mostra che la cifra 2 è presente anche nel codice segreto ma in un posto diverso

Può accadere, ma è un'ipotesi praticamente al limite dell'utopia, che tu indovini al primo tentativo la combinazione esatta.

Ma che logica e astuzia, sarebbe pura fortuna!!

Purtroppo la scritta "È IL NUMERO GIUSTO" compare di solito solo dopo diverse manche, intervallate dalla scritta "RIPROVI?".

La tua "grande mente" dovrà prendere atto ad ogni tentativo di tutte le informazioni ricevute in precedenza per ricostruire alla fine il numero segreto.

Dato che per passare alla prova successiva devi ogni volta premere EXE,

hai a disposizione prima di continuare tutto il tempo che ti occorre per riflettere e ragionare attentamente.

Il gioco si presenta più semplice se ti avvali di carta e penna per riportare tutte le scelte da te effettuate e le informazioni fornite dal computer; in questo modo hai costantemente sott'occhio l'intero quadro e di sicuro arriverai alla soluzione in un numero minore di tentativi. Dopo tutto anche nel vero Master Mind si ha sempre davanti la tavola di decifrazione, contenente i pioli collocati ad ogni manche.

Se vuoi però accettare un nostro consiglio, prova a indovinare la combinazione senza alcun aiuto, facendo ricorso unicamente alle tue capacità mnemoniche e vedrai che, con un po' di concentrazione, tutte le varie fasi del gioco, anche se non riportate sulla carta, saranno ben presenti nella tua mente. Dopo un po' di esercizio tre o quattro prove ti saranno sufficienti per riprodurre esattamente il codice segreto.

Questo gioco è disponibile anche in compagnia: tu ed i tuoi amici vi sottoporrete uno alla volta alla sfida lanciata dal computer e sarà proclamato vincitore colui che ha usato il minor numero di tentativi di decifrazione.

IL LISTATO

```
  5   PRINT "MASTER**MIND"
  6   VAC
 10   $="123456"
 20   FOR I=1 TO 4
 30   N(I)=INT(RAN#*6)+1
 40   FOR Z=1 TO I
 50   IF N(I)=N(Z-1); GOTO 30
 60   NEXT Z
 70   A$(I)=MID(N(I),1)
 80   NEXT I
100   INPUT "ORA IO PENSO UN NUMERO***********SAI
      INDOVINARLO", $
105   W=W+1
108   IF $=A$(1)+A$(2)+A$(3)+A$(4) THEN 180
110   FOR I=1 TO 4
120   IF A$(I)=MID(I,1); H$=H$+"■": NEXT I: GOTO 160
130   FOR Z=1 TO 4
140   IF A$(I)=MID(Z,1); J$=J$+"□": NEXT I: GOTO 160
150   NEXT Z: NEXT I
160   PRINT $; CSR 8; H$; J$
```

```
170   J$="": H$="": INPUT "ƀƀ RIPROVI", $: GOTO 105
180   PRINT $; "ƀE' IL NUMERO GIUSTOƀƀƀƀ MOSSE:"; W
```

Il segno ■ nella riga 120 è fatto con MODE . SHIFT Z
Il segno □ nella riga 140 è fatto con MODE . SHIFT S

(restano 152 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A$(1)...A$(4)	cifre della combinazione scelta del CASIO
H$	insieme dei quadrati neri
J$	insieme dei quadrati bianchi
W	numero dei tuoi tentativi
$	contiene dapprima la sequenza "123456" e in seguito la combinazione da te impostata

Osserviamo ora il listato di questo Master Mind al computer.

Esso si presenta piuttosto semplice e probabilmente potresti comprenderlo senza bisogno di alcun chiarimento. Se però sei all'inizio della tua strada verso la programmazione forse un paio di cosette avranno interrotto la tua lettura: vediamole quindi assieme.

Le prime linee del programma contengono le istruzioni necessarie al computer per selezionare le quattro cifre che costituiscono il codice segreto.

Ti sarai forse chiesto per quale motivo la scelta di questi numeri non è avvenuta semplicemente facendo uso per quattro volte della classica funzione RAN#, ma tramite un meccanismo un po' più complesso. Infatti vengono sì selezionati quattro numeri casuali (linea 30) ma essi sono poi usati per estrarre, tramite la funzione MID, altrettanti caratteri corrispondenti dalla variabile $ che, come puoi osservare, è "123456".

In questo modo le quattro cifre scelte non esprimono più valori numerici, ma sono soltanto caratteri che entrano a far parte del vettore A$() (linea70).
Risulterà così più semplice in seguito operare un controllo tra le cifre "pensate" dal computer e quelle inserite da te, in quanto anch'esse considerate semplici caratteri della variabile $.

Essa cessa quindi di contenere la sequenza 1,2,...6 per rappresentare, a questo punto, la tua combinazione (linea 100).
Potresti obiettare che l'input dei dati poteva avvenire anche tramite una variabile numerica. Hai perfettamente ragione, ma tieni presente che sarebbe stato molto meno semplice indicare la singola cifra del numero da te impostato per poterla confrontare con quelle della sequenza nascosta.

Altre linee il cui significato può non esserti immediatamente chiaro sono quelle in cui il CASIO evita la scelta di un numero già estratto.

Ciò avviene nelle righe 40,50,60 dove ogni cifra, prima di essere considerata elemento di A$, viene confrontata con tutte le precedenti. Così ad esempio N(4), tramite il ciclo su Z, sarà comparata con N(3), N(2), N(1) e solo nel caso risultino tutte diverse si proseguirà nel programma; altrimenti si ritornerà alla linea 30.
L'ultimo problema riguarda la tecnica con cui, ad ogni tuo tentativo, il computer fornisce le informazioni che ti aiuteranno a svelare la sua combinazione.

Si è dovuto fare in modo che comparissero dapprima tutti i quadrati neri e successivamente quelli bianchi; questo per evitare che una perfetta corrispondenza cifra-simbolo offrisse informazioni non previste.

Infatti ad esempio una risposta del tipo

2 4 6 5 → ■ □ ■ □

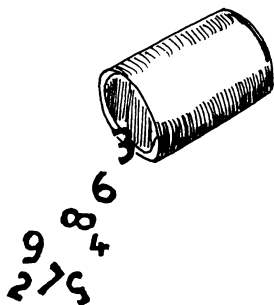
ti avrebbe immediatamente consentito di giungere alla soluzione 2 5 6 4. Nella linea 120 il computer calcola il numero dei quadrati neri da visualizzare, iniziando a controllare il primo carattere da te digitato ed il corrispondente nel codice segreto.

In caso di risposta affermativa si continua esaminando le seconde cifre delle due combinazioni e così via, altrimenti si procederà alla ricerca dei quadrati bianchi.

Ciò viene effettuato nelle linee 130,140 dove, tramite un ciclo su Z interno a quello su I, ogni carattere del tuo tentativo viene confrontato con tutti quelli della sequenza nascosta.

A$(1) con MID(1,1) MID(2,1) MID(3,1) MID(4,1) ecc.

Potrà apparire inutile comparare nuovamente le cifre che occupano la stessa posizione, operazione già compiuta nella precedente ricerca dei "■", ma ciò non causa inconvenienti e in più permette di semplificare notevolmente l'algoritmo relativo.

REVERSE

Non sempre complessità è sinonimo di difficoltà. Talvolta un gioco strutturato in modo semplice risulta meno immediatamente risolvibile e più divertente di uno estremamente sofisticato.

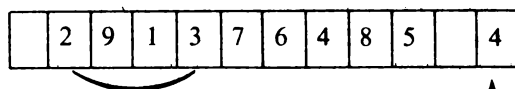
Potrebbe essere il caso di questo passatempo, non per nulla molto diffuso. Non lasciarti ingannare dal suo aspetto dimesso: un antico proverbio dice che l'abito non fa il monaco e, sempre restando in tema, vedrai che, come per le ciliegie, una prova tira l'altra.

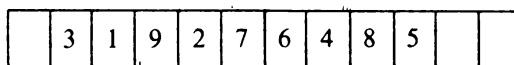
....COME SI GIOCA

Dovrai cercare di disporre in ordine crescente, da sinistra a destra, le cifre dall'uno al nove inizialmente visualizzate in modo casuale. Per fare ciò ti occorreranno più tentativi e ogni tua mossa consisterà nell'indicare quante cifre, partendo da sinistra, intendi invertire rispetto alla loro reciproca posizione. Ma passiamo ad un esempio pratico osservando ciò che compare sul display.

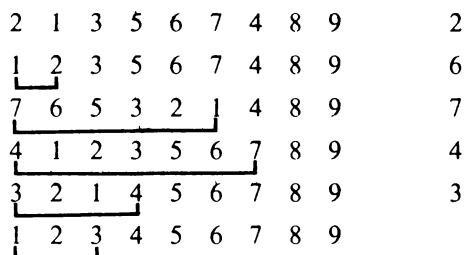
Dopo la lettura del record, posto arbitrariamente a 12 e che tu certamente

supererai in breve tempo, premendo EXE compariranno in successione le cifre da disporre nel loro ordine naturale, che resteranno visualizzate occupando quasi tutto il quadro.

A questo punto tu, al termine delle necessarie elucubrazioni mentali, dovrai inserire una cifra dall'1 al 9 che occuperà l'ultimo spazio alla destra del display

In seguito alla scelta effettuata apparirà subito la nuova situazione

	3	1	9	2	7	6	4	8	5		

(in questo caso con le prime quattro cifre invertite) e la possibilità di spremerti le meningi per la prossima mossa.

Un esempio completo di soluzione è il seguente

2	1	3	5	6	7	4	8	9	2
1	2	3	5	6	7	4	8	9	6
7	6	5	3	2	1	4	8	9	7
4	1	2	3	5	6	7	8	9	4
3	2	1	4	5	6	7	8	9	3
1	2	3	4	5	6	7	8	9	

ed in sole 5 mosse avresti risolto il gioco.

Al termine della prova comparirà il numero dei tentativi che ti sono stati necessari per ristabilire l'ordine dovuto e, nel caso sia stato migliorato il best-score, il record viene aggiornato; a questo punto non avrai che da premere EXE per dare vita alla partita successiva.

Non pensare che la migliore soluzione sia sempre quella più scontata ed evidente. Talvolta una sequenza meno immediata ti sarà più vantaggiosa, eventualità ovviamente controbilanciata dal maggior rischio di commettere errori di valutazione.

IL LISTATO

```
  1    PRINT "* ƀ REVERSE ƀ*"
  5    A=12
  8    PRINT " ƀ RECORD="; A: N=0: X=0
 10    $="123456789"
 20    FOR M=1 TO 9
 30    N(M)=INT(RAN#*9)+1
 32    FOR Z=1 TO M
 35    IF N(M)=N(Z-1); GOTO 30
 38    NEXT Z
 40    A$(M)=MID(N(M),1)
 50    PRINT CSR M; A$(M);
 60    NEXT M
 70    L$=KEY: IF L$="" THEN 70
 80    K=VAL(L$): PRINT CSR 11; L$;: X=X+1
 85    N=INT(K/2)
 90    FOR M=1 TO N
100    Y$=A$(M)
110    A$(M)=A$(K+1-M)
120    A$(K+1-M)=Y$
130    NEXT M
140    FOR M=1 TO K
150    PRINT CSR M; A$(M);
160    NEXT M
170    FOR M=1 TO 9
180    IF A$(M)≠MID(M,1) THEN 70
190    NEXT M
195    FOR M=1 TO 200: NEXT M
200    PRINT: PRINT "OKAY! ƀ ƀ ƀ ƀ"; X; " ƀ MOSSE ƀ ƀ"
210    IF X<A; A=X
220    GOTO 8
```

(restano 158 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A	record della seduta di gioco
A$(1)...A$(9)	conversione in carattere di N(1)..N(9)
K	la tua mossa
N	numero degli scambi da effettuare
N(1)....N(9)	disposizione casuale delle 9 cifre
X	il tuo punteggio
Y$	registro intermedio per lo scambio di due elementi

Come avrai potuto notare il programma non è concettualmente complesso e questa sua linearità si rispecchia nell'osservazione del listato, sufficientemente chiaro (almeno così riteniamo) una volta compresa la logica dell'algoritmo. Pur tuttavia pensiamo possa essere utile chiarire un paio di argomenti meritevoli di attenzione: ma passiamo subito al dettaglio.

Con il ciclo su M che ha inizio alla linea 20 viene stabilita e visualizzata la configurazione di partenza con una tecnica già vista (WARGAME, MASTER MIND) che consiste nell'estrarre caratteri dalla variabile esclusiva $. Analogamente a quanto avveniva in WARGAME la "complicazione" che ne deriva si è resa necessaria per evitare che la stampa delle cifre fosse ottenuta con uno spazio di separazione (riservato al segno) fra una e l'altra.

Il ciclo interno su Z (linee 32/38) ha lo scopo di evitare che venga scelta una cifra già selezionata e a questo proposito, durante la prima stesura del programma, si verificava un fatto curioso. La stampa dei primi otto caratteri, memorizzati nel vettore A$(), avveniva regolarmente, ma per quanto riguardava il nono ed ultimo l'attesa era spesso vana. Il programma entrava in loop e non c'era verso di uscirne. Cosa stava succedendo?

A volte è sufficiente una piccola disattenzione per determinare il non perfetto funzionamento del nostro lavoro. Infatti per evitare la ripetizione delle cifre ognuna di esse veniva confrontata con tutte le precedenti estrazioni e, per semplificare il listato, il confronto (linea 35) era effettuato anche con un inutile N, più precisamente con N(0) (quando Z assume il valore 1).

Nulla da eccepire sul piccolo trucchetto, salvo il fatto che la stessa variabile N era usata all'interno del programma (linea 85) e ad ogni prova restava ovviamente memorizzato l'ultimo valore da essa assunto.

Per ovviare alla cosa molte erano le soluzioni possibili: cambiare variabile, modificare il ciclo su Z oppure, come è stato fatto, aggiungere l'istruzione N=0 all'inizio del programma.

Bene, detto di questo piccolo contrattempo, incerto del mestiere, soffermiamoci un poco sul gruppo di istruzioni (linee 85/130) che determinano, in seguito alla cifra inserita, il passaggio dalla vecchia alla nuova situazione.

Come spesso accade l'algoritmo è più facile a farsi che a dirsi e forse sarai agevolato se ti presentiamo un caso concreto.

Supponiamo che la combinazione da riordinare sia

3 2 7 8 9 1 4 6 5

e che tu decida di invertire le prime cinque cifre.

Evidentemente il 9 dovrà scambiarsi di posto con il 3, e analogamente accadrà per il 2 e l'8, mentre il 7 è destinato a rimanere dov'è.

9 8 7 2 3 1 4 6 5

Avrai certo compreso che, anche se tu hai selezionato un 5, in realtà sono soltanto due gli spostamenti da effettuarsi, e da questa osservazione deriva il nostro algoritmo.

La cifra da te impostata viene divisa per due (linea 85) e la sua parte intera indicherà il numero degli scambi da attuare, compito svolto dal successivo ciclo su M.

Per invertire di posto due elementi, forse lo saprai già, è necessario poter disporre di una terza variabile, che consenta il salvataggio di uno dei due valori prima dello scambio effettivo.

Se facciamo riferimento all'esempio precedente, la sequenza che inverte i primi due numeri è:

3 = A$(1) viene memorizzato in Y$
9 = A$(5) viene memorizzato in A$(1)
Y$ viene memorizzato in A$(5)

In modo simile si procederà con i successivi numeri e il piccolo calcolo K + 1 - M serve appunto alla determinazione dell'elemento da scambiare con quello di posto M.

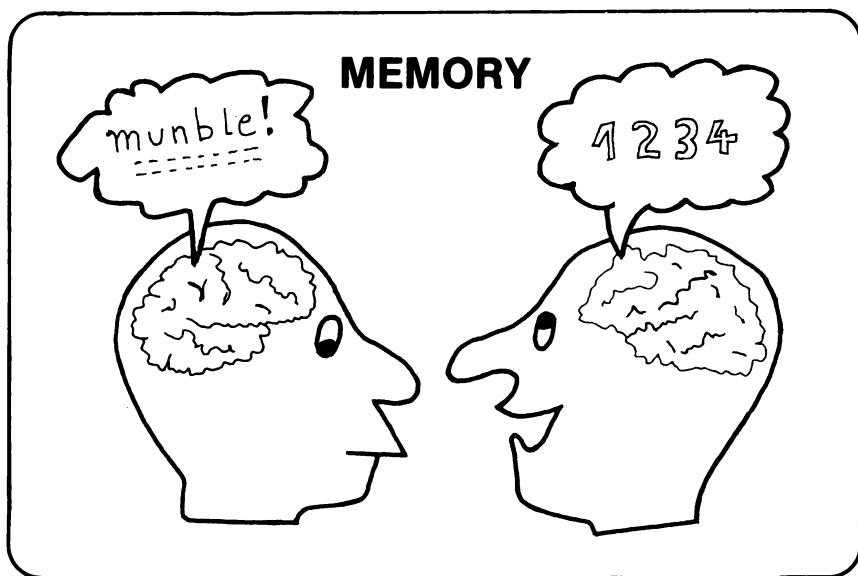
Potrebbe forse essere utile, per fare cose analoghe in programmi di tua creazione, capire bene come si possono ottenere formule simili.

Per ogni elemento da invertire (con posto M nel vettore) l'indice che ne definisce la destinazione sarà tanto maggiore quante sono le cifre indicate (K) e tanto minore quanto è grande M stesso, per cui ecco spiegati i segni +K e -M della formula. Fissiamo ora la nostra attenzione su un determinato elemento della serie visualizzata, ad esempio il primo: se K=5 esso dovrà trasferirsi al quinto posto, e dunque

K-M + qualchecosa dovrà essere uguale a 5

Sostituendo i valori di K (=5) ed M (=1) si ottiene

5-1+qualchecosa = 5

che fornisce qualchecosa = 1 e quindi la versione definitiva del calcolo (K-M+1).

Ecco ora un modo divertente per conoscere e misurare le capacità "visive" della tua memoria.

Infatti, oltre che un vero e proprio gioco, "Memory" può essere considerato quasi un test da eseguirsi in compagnia dove tu ed i tuoi amici sarete chiamati a darvi battaglia nel campo della percezione. Mi raccomando, non cercare ad ogni costo un avversario scarso; dopo tutto, anche se fai una figuraccia, puoi sempre sostenere che i test psicologici hanno una ben scarsa attendibilità.

....COME SI GIOCA

La dinamica di questo gioco, che richiede quell'agilità mentale comunemente denominata "colpo d'occhio", consiste nel ricordare e ripetere un determinato numero casuale sottoposto alla tua attenzione soltanto per una frazione di secondo.

Questa specie di test è realizzato sotto forma di una gara tra due persone che, dopo l'inserimento dei rispettivi nomi (massimo di 7 lettere), dovranno indicare a quale livello di difficoltà intendono cimentarsi.

La "forza" del gioco varia da 1 a 3 e per le prime volte sarà consigliabile selezionare la n.1, che è la più semplice. Essa consente infatti di osservare il numero visualizzato per qualche attimo in più rispetto ai livelli maggiori; forse potrà sembrarti una differenza di poco conto ma che al lato pratico risulta determinante ed evidentissima. Per quanto ci riguarda, ad esempio, mentre a FORZA 1 siamo riusciti talvolta a memorizzare una sequenza anche di dieci cifre, al terzo livello ciò non è stato più possibile.

Da quanto detto avrai certo capito che il numero proposto non ha sempre la solita lunghezza, ma sarai tu stesso a decidere con quante cifre intendi competere, rispondendo adeguatamente alla relativa domanda.

La partita si compone di 10 prove per ognuno dei due giocatori che a turno saranno appunto chiamati a trascrivere esattamente i caratteri numerici comparsi per un attimo sul display.

È ovvio che i punti acquisiti ad ogni manche sono in relazione alla lunghezza della sequenza ricordata. Più esattamente, in caso di risposta corretta ti verrà attribuito un punteggio pari al quadrato del numero di cifre da te richiesto (il cui massimo è stato fissato a 10). Questo rappresenta una specie di invito al rischio sempre che, naturalmente, non si ecceda nel valutare le proprie capacità.

Talvolta potrà capitarti che la prima cifra scelta dal computer sia uno 0. Ciò costituisce una bella fortuna: infatti, non essendo visualizzata, il numero sarà più corto di quanto a regola avresti dovuto aspettarti, mentre il punteggio non subirà diminuzioni di sorta.

L'opportunità data ai due giocatori di stabilire ad ogni turno quella che si può considerare una scommessa sulla propria bravura rende senz'altro la prova più movimentata, mettendone in risalto anche l'aspetto agonistico. Entrambi potranno di volta in volta, in relazione al punteggio, adottare una diversa strategia di gioco, aumentando la posta se in svantaggio o al contrario limitandosi a richiedere tre o quattro cifre in caso di posizione nettamente sicura.

Il secondo giocatore è comunque leggermente avvantaggiato, potendo fare la sua corsa avendo come riferimento quella dell'avversario.

Per avere sempre un'esatta conoscenza della situazione puoi, quando lo ritieni opportuno, premere il tasto "0" in risposta alla domanda "Quante Cifre?". Vedrai così visualizzarsi sul display, per ogni giocatore, i punti acquisiti e le prove effettuate fino a quel momento.

T	=		2	5			P	=		1	

IL LISTATO

```
  10  VAC
  20  INPUT "NOMI",B$,C$
  30  INPUT "FORZA 1,2,3",A
  40  FOR D=1 TO 10
  50  FOR E=1 TO 2
  60  GOSUB 1000: H(E)=H(E)+K
  70  NEXT E: NEXT D
  90  IF I=J; PRINT "PAREGGIO"; I; "ƀƀAƀ"; I: END
 100  IF I>J; $=B$: F=I: L=J: GOTO 120
 110  $=C$: F=J: L=I
 120  PRINT "HA VINTOƀ"; $;" ƀƀPER"; F; " ƀƀAƀ"; L: END
1000  K=0: PRINT: PRINT A$(E); ",QuanteCifre";
1020  INPUT M: N=0: IF M>10 THEN 1000
1030  IF M=0 THEN 2000
1040  FOR L=1 TO M
1050  N(L)=INT(RAN#*10)
1060  N=N+N(L)*10↑(L-1)
1070  NEXT L: PRINT N;
1090  FOR L=1 TO (4-A)*200: NEXT L
1100  PRINT: INPUT L
1110  IF L≠N; PRINT "ƀƀƀNOO!ƀƀƀ": GOTO 1130
1120  K=M*M: PRINT "ƀƀƀESATTO!ƀƀƀƀƀPUNTI"; K; "ƀƀ"
1130  F(E)=F(E)+1: RETURN
2000  FOR F=1 TO 2
2010  PRINT: $=A$(F): PRINT CSR (12-LEN($))/2; $;: GOSUB 3000
2020  PRINT: PRINT "T="; H(F); "ƀƀP="; F(F);: GOSUB 3000
2030  NEXT F: GOTO 1000
3000  FOR M=1 TO 300: NEXT M: RETURN
```

(restano 19 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A	difficoltà del gioco
B$,C$	nomi dei giocatori
F(1),F(2)	prove disputate
I,J	punteggio dei giocatori
L	il tuo tentativo
N	numero da ricordare
N(L)	singole cifre che compongono N

Praticamente il programma consiste in un'unica e corposa subroutine che ha inizio alla linea 1000 e nella quale sono svolte tutte le principali operazioni dell'algoritmo.

Dato che la variabile M indica il numero delle cifre che vuoi mettere in gioco, tramite il ciclo su L (linea 1040) si ottiene innanzitutto la scelta casuale della combinazione da ricordare e ripetere. Si è questa volta deciso, piuttosto che convertire tali cifre in caratteri, di mantenerne le proprietà numeriche; potrai infatti notare che la stampa inizia da CSR 1, essendo il primo spazio riservato al segno.

Affinchè la sequenza possa essere visualizzata sotto forma di un unico valore aritmetico

abbiamo moltiplicato ogni cifra per potenze crescenti di 10, determinandone così la rispettiva posizione (unità, decine, centinaia..) nel numero definitivo. Ciò viene realizzato con la formula presente nella linea 1060. Ad esempio con un numero di due cifre, poniamo 37, la prima (N(1)) verrà moltiplicata per 1 (10 ↑ 0), mentre la seconda per 10 (10 ↑ 1). La loro somma 7+30 darà appunto come risultato 37.

Il tempo per cui il numero scelto rimane sul display è calcolato e fatto trascorrere nella linea 1090. È ovvio che con una minore difficoltà del gioco

(A) si avrà una prolungata permanenza nel visore e viceversa, secondo la seguente relazione

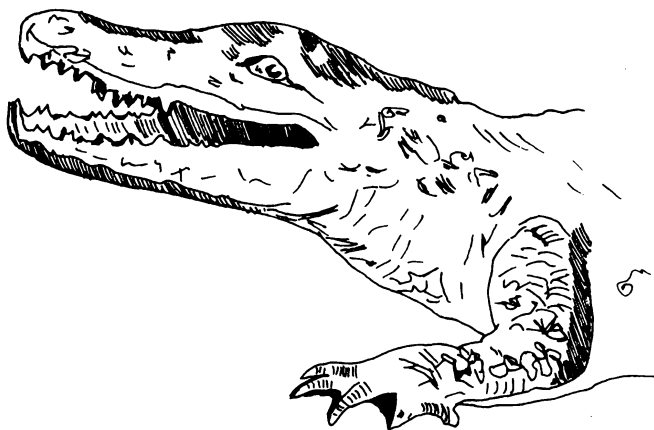
FORZA	TEMPO
1	600
2	400
3	200

La maggiore difficoltà nel comprendere il listato è forse l'utilizzazione di tre vettori nei quali sono memorizzati i nomi dei giocatori, il loro punteggio e le prove già disputate.

	NOME	PUNTEGGIO	PROVE
1° GIOCATORE	A$(1)=B$	H(1)=I	F(1)=G
2° GIOCATORE	A$(2)=C$	H(2)=J	F(2)=H

Una volta che ti saranno chiare le corrispondenze riportate potrai capire più agevolmente lo sviluppo del programma.

Concludiamo con una piccola raffinatezza. Immettendo 0 alla richiesta delle cifre si ha un rimando alla linea 2000, dove avviene la stampa della situazione attuale. Il piccolo calcolo della linea 2010 [PRINT CSR (12-LEN($))/2] ha il solo scopo di centrare nel display il nome del giocatore in relazione alla sua lunghezza. Ad esempio il nome SANDRO (6 lettere) verrà stampato in CSR (12-6)/2 = CSR 3

IL NAUFRAGO

Il grande e misterioso Mississippi, pieno di insidie e pericoli, ti attende. Fai parte di una spedizione scientifica che, partendo dalle sorgenti, si avventura lungo il corso del famoso fiume americano.

Particolare attenzione dovrai prestare ai coccodrilli che infestano queste acque, sempre pronti ad assalire chiunque capiti a tiro dei loro denti aguzzi. Infatti ben presto ti troverai alle prese con uno di questi famelici rettili dal quale potrai salvarti soltanto grazie alle tue ottime qualità di nuotatore.

....COME SI GIOCA

La barca simbolizzata con il segno % attraversa lo specchio d'acqua visibile e ad un certo punto, probabilmente a causa di un'errata manovra, cadi e ti ritrovi nelle acque fluviali. Mentre tu (.) stai ancora riprendendoti dallo spavento, il battello prosegue la sua corsa; poi, accorgendosi della tua scomparsa, si ferma ad attenderti.

Ma ecco che all'improvviso compare un terribile coccodrillo (>) che aprendo e chiudendo le fauci si dirige inesorabilmente verso di te.

Per raggiungere la barca dovrai premere, ripetutamente e il più velocemente possibile, le varie cifre dallo 0 al 9 fino ad indovinare quella "pensata" dal computer in quel momento. Ogni volta che riuscirai nell'intento avanzerai di uno spazio e diminuirà così la distanza che ti separa dai tuoi compagni.

Come avrai capito è un passatempo nel quale la fortuna ha un ruolo determinante e che potrai giocare come solitario nei momenti di relax.

Infatti avendo dato risalto alla parte grafica risulta abbastanza divertente seguire i movimenti del naufrago e soprattutto dell'alligatore del quale risulta ben simulato l'aprirsi e il chiudersi della bocca e l'eventuale pranzetto.

IL LISTATO

```
  1   PRINT "IL NAUFRAGO"
 10   FOR I=11 TO 0 STEP -1: PRINT CSR I; "%";
 20   IF I=6; PRINT CSR 7; ".";
 25   GOSUB 500: PRINT CSR I; " ƀ";: NEXT I: PRINT CSR 0; "%";
 30   GOSUB 500: I=12: G=7
 40   I=I-1: IF I=G THEN 80
 50   FOR K=1 TO 9
 60   PRINT CSR I; ">";: GOSUB 700: PRINT CSR I; "=";: GOSUB 700
 70   NEXT K: PRINT CSR I; " ƀ";: GOTO 40
 80   FOR K=1 TO 40: PRINT CSR I; "."; CSR I; ">";: NEXT K
 90   PRINT: PRINT CSR 4; "SLURP": GOTO 10
150   GOSUB 500: PRINT: PRINT "SEI IN SALVO": GOTO 10
500   FOR J=1 TO 100: NEXT J: RETURN
700   C=INT(RAN#*10): E$=KEY
710   IF E$≠""; IF E$≥"0"; IF E$≤"9" THEN 730
720   GOTO 760
730   V=VAL(E$): IF V≠C THEN 760
740   G=G-1: PRINT CSR G; ". ƀ";
750   IF G=0; PRINT CSR 0; "%";: GOTO 150
760   FOR J=1 TO 15: NEXT J: RETURN
```

Il segno % nelle linee 10, 25, 750 è fatto con MODE . SHIFT Q
Il segno . nelle linee 20, 80, 740 è fatto con MODE . SHIFT X

(restano 153 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
C	numero da indovinare
G	posizione del naufrago
I	posizione della barca e successivamente del coccodrillo

Grazie alla grafica del CASIO è possibile creare sul display l'illusione di movimento, la così detta "animazione" che può rendere taluni giochi più divertenti e simpatici e questo programma si presta particolarmente ad osservazioni sull'argomento.

Diversi sono i caratteri che qui appaiono in "azione" e che rappresentano la barca, il naufrago e il coccodrillo.

Il movimento della barca (%) simulato nelle linee 10/25 non dovrebbe creare troppe difficoltà se avrai ben compreso il meccanismo necessario usato in precedenza nel programma Rolling. Si tratta di:

a) stampa del carattere corrispondente
b) perdita di tempo (subroutine 500)
c) successiva cancellazione tramite la stampa di uno spazio
d) visualizzazione del carattere nella nuova posizione.

Anche gli spostamenti del naufrago sono stati ottenuti con la solita tecnica; in questo caso però non si è resa necessaria la perdita di tempo tra le diverse posizioni del carattere, in quanto queste non variano fino a che il giocatore non avrà indovinato il numero "segreto". Ciò avviene quando la cifra scelta casualmente nella linea 70 (C) risulta uguale a quella premuta sulla tastiera (V).

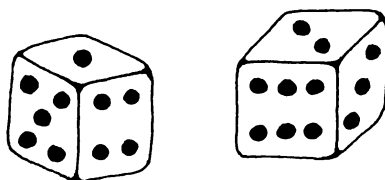
In assenza di un rimando alla subroutine 500 è stato possibile compattare sotto un'unica istruzione (PRINT CSR G; ".ƀ") le due fasi di "pulizia" e successiva stampa.

Naturalmente, essendo il naufrago diretto verso la sinistra del display, la nuova visualizzazione sarà preceduta da un decremento (G=G-1) dello spazio di stampa e l'annullamento della "vecchia" posizione si ottiene giustapponendo uno spazio al carattere che rappresenta l'omino (.ƀ).

Una interessante possibilità per evitare l'eventuale "tilt" del gioco in seguito ad errate pressioni sulla tastiera è quella sfruttata nella linea 710.

L'uso dei tre IF consecutivi permette di controllare (alla linea 730) se la cifra è stata indovinata, soltanto nel caso si prema effettivamente un numero compreso tra 0 e 9 e non, ad esempio, una lettera. Infatti, come ben saprai, ad ogni carattere è associato un particolare codice (ASCII) che consente l'ordinamento di ogni simbolo. Ciò permette in questo caso di selezionare soltanto i caratteri con codice maggiore o uguale a quello di zero (IF E$⩾"0") e minore o uguale a quello di 9 (IF E$⩽"9") che corrispondono appunto a quelli delle cifre comprese fra 0 e 9.

Passiamo ora alle considerazioni riguardanti l'animazione del coccodrillo. Questo, oltre a scorrere lungo il display, apre e chiude le fauci. Tale illusione di movimento (linee 50/60/70) si può così schematizzare:

e quindi ripetizione per 9 volte delle quattro fasi del ciclo.

Dopo giochi di memoria, abilità e prontezza questo nostro YAZZY impegnerà invece le tue capacità riflessive e strategiche, non escludendo però il fattore fortuna.

Non avrai come attento avversario il tuo CASIO, ma questa volta il computer sarà solo un tramite che permetterà a te e ad un amico di gareggiare in tale rilassante passatempo che richiede anche l'uso di carta e penna.

Leggi ora attentamente il paragrafo che segue poichè la corretta visione delle regole del gioco è indispensabile ad una buona riuscita dello stesso.

....COME SI GIOCA

Forse conoscerai già il YAZZY: è un gioco d'azzardo che consiste nel lanciare contemporaneamente cinque dadi, sulle cui facce sono stampate le carte da Poker. Attraverso questi tiri devi cercare di formare determinate combinazioni vincenti, ognuna delle quali fornisce un diverso punteggio.

YAZZY può essere molto efficacemente simulato dal computer che per-

mette infatti di creare le stesse condizioni di casualità e imprevedibilità caratteristiche del lancio dei dadi.

Dopo il titolo appariranno alcune istruzioni che devi ben tenere a mente durante tutta la gara; quindi si visualizzano, in seguito alla pressione del tasto EXE, cinque numeri compresi tra 1 e 6.

Nella nostra versione computerizzata queste cifre indicano il risultato del lancio dei cinque dadi e sostituiscono le figure delle carte che non potevamo rappresentare.

Il primo giocatore, dopo aver osservato i numeri casualmente estratti, considera se ciò che è uscito "gli sta bene" e corrisponde ad una delle possibili combinazioni che potrai tra breve osservare nella tabella che ti presentiamo. In questo caso fortunato non gli resta che segnare nella tabella stessa, riportata su un foglio da ognuno dei due partecipanti, il punteggio effettuato, secondo le modalità che saranno descritte; quindi premendo il tasto "=" passerà il gioco all'avversario.

È possibile, se ritenuto necessario, chiedere per due volte il cambio-carte. Questa operazione si attua digitando le cifre (dall'1 al 5) corrispondenti alla posizione dei numeri di cui si vuole ottenere il rilancio e premendo, dopo l'ultimo inserimento, il tasto "."

In questa situazione, ad esempio, si intende cambiare la prima, la seconda e la quinta cifra (rispettivamente il 3, il 5 e l'1). Dopo la seconda richiesta di cambio-carte il gioco passa automaticamente all'altro giocatore che inizierà la sua manche premendo EXE. Eventuali errori di battitura - piuttosto frequente è ad esempio il caso in cui la cifra venga premuta un po' troppo a lungo e sia quindi visualizzata più volte - possono essere eliminati utilizzando il tasto " E " che, pulendo la parte del display a destra dei "♥♥", la rende libera per una nuova e corretta immissione dati.

Veniamo ora alla parte più importante del gioco che riguarda le combinazioni da formare ed il punteggio.

TABELLA

COMBINAZIONI	PUNTEGGIO
TRIS di 1	
TRIS di 2	
TRIS di 3	
TRIS di 4	
TRIS di 5	
TRIS di 6	
ABBUONO (50 punti)	
COPPIA	
DOPPIA COPPIA	
TRIS	
FULL	
POKER	
SCALA MINIMA (1-2-3-4-5)	
SCALA MASSIMA (2-3-4-5-6)	
CHANCE	
YAZZY (50 punti)	
TOTALE:	

I due giocatori si saranno premuniti di trascrivere (o fotocopiare) questa tabella; per completarla e raggiungere il massimo punteggio possibile hanno a disposizione 15 prove.

Alla fine di ogni manche si devono segnare i punti relativi alla combinazione ottenuta; è un calcolo molto semplice in quanto il punteggio corrisponde esattamente al valore della combinazione stessa. Un tris di 5 darà così 15 punti, da trascriversi nella tabella accanto alla voce "TRIS di 5" oppure vicino a "TRIS".

Un FULL del valore 36366 varrà ovviamente 24 punti e così per tutte le altre sequenze vincenti.

Sarà bene farti notare che non è obbligatorio riportare esattamente la combinazione definitiva e completa ma può essere utilizzata, se la cosa ti conviene, solo una sua parte. Nell'esempio precedente riferito al Full avresti potuto in alternativa segnarti una coppia di tre, o un tris di sei, una coppia di sei o ancora una doppia coppia (3366).

Forse, se non ti sei mai dedicato a questo passatempo, non sai bene cosa sono CHANCE e YAZZY. È semplicissimo. La CHANCE, che ti consigliamo di tenerti buona per le prove conclusive, ha il suo chiaro significato di "ultima possibilità" e consiste nel segnare semplicemente la somma delle cinque cifre visualizzate. Ti sarà quindi chiaro il perchè del nostro consiglio: utilizzala infatti proprio quando non sei riuscito a formare niente di meglio.

Per quanto riguarda il YAZZY, esso si ottiene invece con cinque numeri uguali ed è probabilmente la combinazione più difficile da realizzare, tanto che ti porterà ben 50 punti, indipendentemente dalle cifre che lo compongono.

Devi infine sapere che se riesci a completare tutte le serie dei TRIS, da quello degli uno al sei, ottieni un abbuono di 50, spesso determinante ai fini della vittoria. Per terminare detta serie non occorre però che tu abbia realizzato tutti TRIS effettivi, ti basta totalizzare almeno 63 punti.

Ad esempio se il computer ti assegna una coppia di quattro, puoi benissimo segnarti 8 nella linea della coppia, se questa è ancora inutilizzata, ma potresti anche riportarli nella riga del "TRIS di 4". Per ottenere però l'abbuono avrai in questo caso da recuperare i quattro punti mancanti; ecco che, al primo POKER di 5 che ti uscirà, segnerai 20 nella linea del "TRIS di 5". Ti sarai a questo punto forse chiesto che cosa succede quando, nonostante i due possibili cambi di carte, non si riesca a formare nessuna delle sequenze descritte nella tabella, oppure se ne ottenga una già esaurita avendo anche già usufruito della CHANCE.

Ebbene, non ti resta che scartare una delle combinazioni ancora inutilizzate, che non potrai quindi più ritentare, con una evidente e notevole perdita di punti.

Ci pare ora di aver detto tutto ciò che ti serve per iniziare a giocare. Le regole ti sembrano troppe e un po' complicate? Ma non sai che, come in tutte le cose, a giocare si impara solo giocando? E allora buona fortuna!

IL LISTATO

```
  1  PRINT " ƀƀƀ YAZZY"
  2  PRINT "CON < = > PASSI IL GIOCO ƀƀ";
  3  PRINT " ƀƀ CON <.> CAMBI LE CIFRE ƀƀƀ";
  4  PRINT " ƀ CON < E > CORREGGI GLI ERRORI ƀ"
  5  VAC
 10  $="123456": FOR I=1 TO 5
 20  A=INT(RAN#*6)+1
 30  A$(I)=MID(A,1)
 40  X$=X$+A$(I): NEXT I
 60  PRINT X$; " ♥ ♥ ";
 65  H=6: Z$=""
 70  L$=KEY: IF L$="" THEN 70
 72  IF L$="." THEN 130
 73  IF L$=" E "; PRINT: GOTO 60
 74  IF L$=" = " THEN 400
 75  H=H+1: IF H>11; H=11
 90  IF L$≥"1"; IF L$≤ "5"; PRINT CSR H; L$;: GOTO 120
100  PRINT: GOTO 60
120  Z$=Z$+L$: GOTO 70
130  G=LEN(Z$)
140  FOR I=1 TO G
150  K=INT(RAN#*6)+1
160  M$=MID(K,1)
170  $=Z$
180  W$=MID(I,1)
190  N=VAL(W$)
200  PRINT CSR N-1; M$;
202  A$(N)=M$
205  $="123456"
210  NEXT I
215  X$="": FOR I=1 TO 5
220  X$=X$+A$(I): NEXT I
230  PRINT CSR 7; " ƀƀƀƀƀ";:J=J+1
240  IF J≥2 THEN 400
250  GOTO 65
400  PRINT CSR 7; "PASSA": GOTO 5
```

Il segno ♥ nella riga 60 è fatto con MODE . SHIFT J

(restano 7 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A$(1)...A$(5)	le singole cifre estratte casualmente da $
L$	singolo carattere implementato tramite la funzione KEY
M$	nuova cifra casuale che sostituisce la precedente
N	diminuito di un'unità fornisce la posizione di stampa del nuovo carattere
X$	l'intera combinazione visualizzata
Z$	intera stringa digitata contenente la posizione delle cifre da cambiare

L'algoritmo usato per la realizzazione di YAZZY è piuttosto semplice e presenta spiccate ed ovvie somiglianze con quello di altri giochi dello stesso tipo, come Master Mind, Slot Machine, Memory; tutti giochi nei quali si devono ottenere determinate combinazioni che consentono la vittoria.

La struttura di tali programmi comprende quindi:

— una prima fase in cui il computer genera una certa sequenza di caratteri
— seguono i tentativi del giocatore di formare una determinata combinazione o di indovinare quella "ideata" dal calcolatore
— ed infine la fase di controllo, attraverso la quale il computer analizza le risposte del giocatore per verificarne l'eventuale esattezza, assegnando il relativo punteggio.

Questo schema di massima subisce poi modifiche, ampliamenti, variazioni al momento della stesura dei singoli giochi.

Nella nostra versione di YAZZY, ad esempio, l'ultima fase citata è del tutto assente, in quanto sono i due giocatori e non il CASIO a dover controllare il tipo di combinazione formata e calcolare i punti totalizzati. Abbiamo dovuto invece sviluppare e dedicarci maggiormente alla fase di immissione dati, la più complessa di questo programma, in quanto è possibile ai due avversari chiedere ad ogni manche per ben due volte il cambio delle carte visualizzate.

Ed è questa parte del listato che vale forse la pena di rivedere assieme.

Nulla di nuovo riguardo alla funzione KEY usata più volte (linee 70/74) affinchè il giocatore possa implementare ciò che desidera, dai vari simboli

come la “1_E” che cancella il display in caso di errore, ai caratteri numerici indicanti la posizione delle cifre che intende modificare, caratteri accumulati poi uno alla volta nella variabile Z$ (linea 120). Ti risulterà così evidente che la lunghezza di tale variabile stringa fornirà al computer l'informazione di quante cifre deve sostituire tramite una nuova scelta casuale. Meno scontato è forse ottenere la stampa delle nuove cifre nella stessa esatta posizione di quelle per così dire “eliminate”.

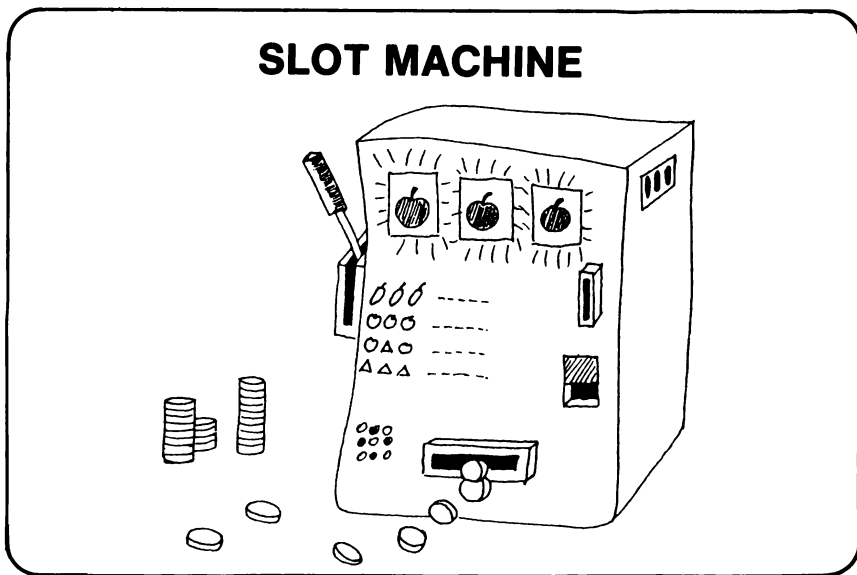
Assegnando il contenuto di Z$ alla variabile esclusiva $, che quindi cessa per il momento di rappresentare le cifre dall'1 al 6, si rende possibile l'uso della funzione MID.

Tramite questa istruzione si estraggono poi uno alla volta i caratteri di $ che indicano, abbiamo già visto, la posizione delle cifre da sostituire (linea 180) e vengono successivamente trasformati in valori numerici (linea 190).

Si otterrà così un numero N che diminuito di un'unità, in quanto la numerazione degli spazi del display inizia da 0, fornisce la locazione in cui la nuova cifra dovrà essere visualizzata. Ma vediamo con un esempio queste operazioni.

Il contenuto della variabile Z$ viene trasferito in $ che risulterà quindi “23”; si estrae da $ il primo carattere che viene convertito in numero (N=2) e la nuova cifra verrà così stampata nello spazio N-1, cioè 1, e andrà a prendere il posto del quattro iniziale, che infatti si voleva sostituire; naturalmente si opererà nello stesso modo per tutti gli altri caratteri presenti in $.

Forse non è il caso di chiarire che, al termine di questa parte del listato, tutte le variabili dovranno essere nuovamente definite: $ tornerà ad essere “123456”, ogni nuova cifra visualizzata prenderà il nome di quella che l'ha preceduta nella medesima posizione (A$()) ed inoltre sarà ulteriormente assegnata a X$ la sequenza finale (linea 220).

SLOT MACHINE

Sei entrato nell'atmosfera luccicante e un po' ovattata di un grande Casinò.
Roulette, Poker, Jack-pot, non sai proprio che cosa scegliere.
Il tuo sguardo corre per tutta la sala ed ecco che vedi, laggiù in fondo, quelle "infernali" macchinette mangiasoldi capaci di lasciare la gente completamente al verde.
Sono le slot-machine; non puoi rifiutare la sfida che esse ti lanciano. Mentre ti avvicini con cautela già vedi mille e mille monete scendere con un assordante ticchettio, tutte per te.
Su, rilassati! Comunque vada, il divertimento è di certo assicurato.

....COME SI GIOCA

Come ben saprai nelle vere slot-machine si possono vedere tre o più simboli diversi che ruotano fino a fermarsi.
Nel caso che la combinazione uscita sia una delle vincenti, si ottiene un premio che è in relazione al tipo di combinazione stessa e all'ammontare della somma giocata.

Anche nella nostra versione queste regole restano valide ed essa non si allontana poi di molto dal gioco che ti può impegnare con le simpaticissime macchinette.

Hai a disposizione un totale di L. 10.000, proprio la cifra equivalente ai gettoni che danno all'entrata del Casinò; ti viene quindi richiesto di scommettere la quota desiderata: inseriscila e premi EXE. Se digiti un numero superiore alla somma che possiedi realmente, per errore o per fare il "furbo", il tuo CASIO, dopo un attimo di esitazione, ti chiede un'altra volta l'input della scommessa.

Vedrai a questo punto visualizzarsi al centro del display tre quadrati neri "■ ■ ■"; tenendo schiacciato un tasto qualsiasi queste caselle inizieranno a lampeggiare e, appena avrai cessato la pressione, compariranno al loro posto tre simboli.

Questi sono stati scelti casualmente dal computer fra i seguenti

e sono il meglio che la grafica del CASIO consenta; per differenziarli anche in base al colore ne abbiamo utilizzati tre bianchi e tre neri. Non hai ora che da verificare se la combinazione uscita è tra le vincenti.
Puoi fare TRIS se i simboli scelti e visualizzati sono tutti uguali, ad esempio:

In questo caso la vincita che ti verrà assegnata è di ben 20 volte la posta: sei davvero fortunato!

Un'altra combinazione vincente è COLORE, se i tre simboli sono tutti bianchi o tutti neri e necessariamente diversi come forma:

Questa vincita vale 10 volte la posta da te giocata.
L'ultimo tipo di combinazione premiata è stato da noi chiamato "COPPIA" e si ha quando i due simboli laterali sono uguali e quello interno è del colore opposto.
Così ○ ♦ ○ o ♣ Ω ♣

ti darà una vincita pari a 5 volte la tua scommessa.

Forse questo schema riassuntivo potrà esserti utile per ricordare meglio

COMBINAZIONE	ESEMPIO	VALORE
TRIS	□ □ □	20 volte la posta
COLORE	○ Ω □	10 volte la posta
COPPIA	○ ♦ ○	5 volte la posta

Sarà ad ogni modo lo stesso computer ad informarti dell'eventuale manche fortunata, sommando al denaro che hai la vincita ottenuta e visualizzando sul pannello la nuova situazione.

Se la dea bendata non ti ha invece sorriso ti verrà ovviamente sottratta la cifra scommessa.

Il gioco continua fino a quando, tra alti e bassi, non avrai perso tutto ciò che possiedi.

Con un minimo di assennatezza, cautela ed astuzia, ma soprattutto con molta fortuna, vedrai che questo slot-game ti impegnerà a lungo, anche se alla fine il crollo è inevitabile.

IL LISTATO

```
  1   PRINT "SLOT-MACHINE"
  5   VAC
 10   L=10000
 15   T=250: IF L=0; GOSUB 999: PRINT: PRINT "FINE HAI PERSO
      TUTTO";: END
 20   GOSUB 999: PRINT: PRINT " ƀHAI"; L; " ƀLIRE ƀ ƀ ƀ ƀ ƀ
      SCOMMESSA";
 30   INPUT P
 40   IF P>L THEN 20
 50   $=" ♠ ♦ ♣ □○Ω"
 60   PRINT CSR 5; "■ ■ ■";
 70   IF KEY="" THEN 70
 80   PRINT CSR 5; " ƀ ƀ ƀ";: T=20: GOSUB 999: PRINT CSR 5;
      "■ ■ ■";
 90   IF KEY≠"" THEN 80
100   FOR B=1 TO 3
110   A=INT(RAN#*6)+1
112   IF A>3; B(B)=1
```

```
120    PRINT CSR 4+B; MID(A,1);: GOSUB 999
130    F(B)=A: NEXT B
140    IF F(1)≠F(3) THEN 400
150    IF B(1)≠B(2); V=5: GOTO 500
160    IF F(1)≠F(2) THEN 490
170    V=20: GOTO 500
400    IF B(1)≠B(3) THEN 490
410    IF B(1)≠B(2) THEN 490
420    IF F(1)=F(2) THEN 490
430    IF F(2)=F(3) THEN 490
440    V=10: GOTO 500
490    V=-1
500    L=L+V*P: B(1)=0: B(2)=0: B(3)=0: GOTO 15
999    FOR Z=1 TO T: NEXT Z: RETURN
```

I segni ♠ ♦ ♣ □ ○ Ω nella riga 50 sono fatti con MODE . SHIFT H, K, L, S, A, N

Il segno ■ nelle righe 60 e 80 è fatto con MODE . SHIFT Z

(restano 54 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A	seleziona il simbolo da visualizzare
B(1)...B(3)	0 od 1 a seconda del colore del simbolo corrispondente
F(1)...F(3)	posizione in $ dei simboli estratti
L	somma a disposizione
P	cifra scommessa
V	coefficiente per cui verrà moltiplicata la somma scommessa

La parte del gioco che ha richiesto maggiore impegno ed una discreta conoscenza delle tecniche di programmazione è quella che permette di verificare se la sequenza uscita è tra quelle vincenti.

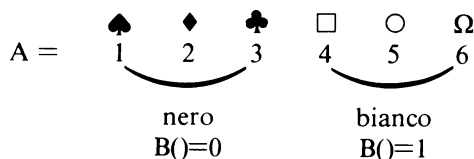
Avrai innanzitutto già osservato che "♠ ♦ ♣ □ ○ Ω" sono stati assegnati alla variabile $ (linea 50) e tre di essi vengono scelti casualmente ed estratti

tramite la funzione MID (linea 120). Ed è a questo punto che inizia il suddetto controllo.

Esso non avviene direttamente tra i simboli, ma si è preferito operare con le cifre. I tre numeri casuali generati dalla funzione RAN# e che indicano i caratteri estratti da $, vengono considerati poi come elementi di un vettore ed annotati come F(1) F(2) F(3), rappresentanti nell'ordine il primo, il secondo e il terzo simbolo della combinazione visualizzata.

Un'altra avvertenza per rendere il controllo più semplice e meno contorto è stata quella di definire un secondo vettore B() di tre componenti, assegnando ad ognuno di essi un diverso valore (0 oppure 1) in relazione al colore del simbolo corrispondente.

Dato che nella variabile $ sono stati collocati dapprima tutti i caratteri neri e poi quelli di colore bianco, ti risulterà chiaro che se il numero selezionato è maggiore di 3 ($A > 3$) sarà stato scelto un simbolo bianco.

In tal caso la variabile B() assumerà valore 1, mentre se $A \leqslant 3$ ed è così stato estratto un carattere nero, B() sarà uguale a 0.

Ecco che la successione B(1) B(2) B(3) indicherà così il colore dei tre segni grafici della combinazione. Ad esempio

$$B(1) = 1$$
$$B(2) = 0$$
$$B(3) = 1$$

ti mostra evidentemente che il primo e il terzo simbolo sono bianchi, mentre quello interno è nero.

In questo modo nel controllo possiamo ora utilizzare solo numeri, sia quelli del vettore F() che indicano la posizione in $ del simbolo estratto, sia lo 0 o l'1 del vettore B() che ne rappresenta il colore.

A questo punto ti sarà facile comprendere che se il primo ed il terzo carattere risultano diversi (linea 140), l'unica possibilità che ti rimane è quella di aver realizzato COLORE, la cui uscita sarà controllata a partire dalla linea 400.

Nel caso contrario (F(1)=F(3)) hai buone possibilità di vincere con il TRIS o almeno con la COPPIA.

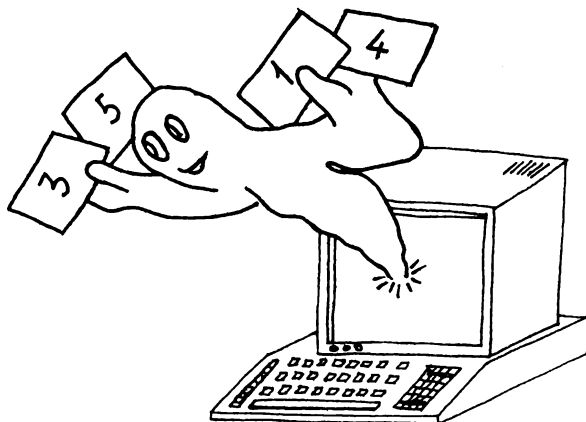
La prima ipotesi dovrebbe comportare innanzitutto la presenza dello stesso colore anche nel simbolo centrale, e tale test viene effettuato nella linea 150. In caso positivo (B(1)=B(2)) dovrà essere ancora verificata l'uguaglianza di F(2) con uno dei caratteri laterali, ad esempio con il primo (linea 160).

Se invece dal controllo della linea 150 è risultato che i colori del primo e del secondo simbolo sono opposti, si passa subito alla linea 500 avendo realizzato necessariamente "COPPIA".

Poichè il programma ha occupato un numero di passi minore di quelli a disposizione, abbiamo potuto rendere il listato più comprensibile, evitando ad esempio di scrivere G, H, I in luogo di F(1) F(2) F(3) e lo stesso dicasi per il vettore B().

Per il solito motivo si è inoltre preferito ripartire le istruzioni relative al controllo della combinazione "COLORE" in ben quattro linee (400 / 430); in presenza di problemi di spazio di memoria saremmo stati costretti a porre i quattro IF consecutivamente nella stessa linea ed ecco come si sarebbe presentata la relativa parte del listato

```
400 IF C=E; IF C=D; IF G≠H; IF H≠I THEN 440
410 GOTO 490
```

IL TAGLIO

Hai dormito bene? Sei completamente rilassato? Hai il bioritmo favorevole? Soltanto se hai risposto per tre volte "si" ti consigliamo di cimentarti in questo gioco. Perdere è sempre spiacevole, ed in modo particolare contro un computer, ma quello che dà veramente fastidio è il perdere con un computer che non bara, che non possiede la strategia vincente e che pure ti supera in media 7 volte su 10 in un gioco basato sulla psicologia, sull'intuito, sull'astuzia. Impossibile?
Forse una spiegazione esiste: che il tuo CASIO legga nel pensiero?

....COME SI GIOCA

Qualche tempo fa, in un numero della rivista "Le Scienze", appariva un articolo in cui veniva presentato questo gioco ideato dallo scrittore stesso, Hofstadter, e da un suo collega.

La gara, per due contendenti, consisteva nello scegliere entrambi un numero intero compreso fra 1 e 5, per poi confrontarli: se la loro differenza era diversa da 1 ciascun giocatore riceveva un punteggio parziale pari alla cifra

indicata; se la differenza era esattamente 1 il giocatore con il numero minore li sommava entrambi, "tagliando" così l'avversario, che rimaneva all'asciutto.

La prova, pur nella sua semplicità di regole, risulta particolarmente accattivante per gli innumerevoli risvolti psicologici ed astuzie che riesce a scatenare nei tentativi di mandare "fuori pista" l'avversario.

In questa versione, dove sei impegnato contro il computer, la gara termina al raggiungimento dei 100 punti; in caso di manche pari (stessa cifra estratta da entrambi) non viene assegnato alcun punteggio. Ad ogni turno comparirà la scritta

IO: ■ TU:?

e dovrai inserire la tua cifra (da 1 a 5) che prenderà il posto del punto interrogativo. È bene farti notare che a questo punto il computer ha già scelto la sua, e quindi in pratica ti viene sempre concesso di "tirare" per secondo, anche se il vantaggio ricavato è ben misero, in quanto il numero scelto dal Casio rimarrà nascosto dietro il ■, ed apparirà soltanto dopo la tua scelta. Dopo ogni manche, salvo nel caso di parità, verrà visualizzato per qualche istante il punteggio accumulato fino a quel momento, in modo da avere la situazione sempre sotto controllo.

In questa stimolante disfida molte sono le possibilità di gioco, dalle più semplici alle più sofisticate, quali ad esempio puntare sempre su numeri bassi, che non verranno quasi mai tagliati, o numeri alti, che ad un elevato contenuto di rischio controbilanciano maggiori punteggi. Senza parlare poi dei possibili schemi di gioco, come potrebbe essere il tirare tre 5 di fila sperando di indurre l'avversario a "tagliare" con un 4 ma fregandolo astutamente con un 3 al momento giusto. Il tutto con un crescendo di livelli di "depistaggio psicologico" (come dice l'autore) che rendono il gioco sicuramente divertente.

Vedi infine di non lasciarti condizionare dagli epiteti che compaiono quando ti fai "tagliare", né dalle esclamazioni in prossimità della conclusione. Potrebbe capitare, anche se raramente, che il computer venda la pelle dell'orso (la tua) prima di averla nel sacco.

IL LISTATO

```
1    VAC
2    $="SCARSOFIGURASMACCOPENA ƀ ƀ"
4    PRINT "IO:■ ƀ ƀ ƀTU:?";: X=INT(RAN#*68)+1
```

```
5    B$=KEY: IF B$="" THEN 5
6    IF B$>"0"; IF B$≤"5"; PRINT CSR 10; B$;: C=VAL(B$):
     GOTO 8
7    GOTO 5
8    Z$="1": IF X>10; Z$="2"
9    IF X>36; Z$="3"
10   IF X>49; Z$="4"
11   IF X>65; Z$="5"
12   H=2↑332: PRINT CSR 3; Z$;: W=VAL(Z$): GOSUB 30
13   IF C=W; PRINT "NESSUN PUNTO";: GOSUB 30: GOTO 4
14   IF ABS(C-W)≠1; N=N+C: M=M+W: GOTO 17
15   IF C<W; N=N+C+W: GOTO 17
16   J=INT(RAN#*4): M=M+C+W: PRINT "CHE b"; MID(J*6+1,6);
     "!";: GOSUB 30
17   PRINT "IO"; M; CSR 7; "TU"; N;: GOSUB 30: IF M<100; IF N<100
     THEN 21
18   IF M>N; PRINT "NON PUOI COMPETERE CON ME!! b b b": END
19   IF M=N; PRINT " b bPARITA'": END
20   PRINT "BRAVO, MI HAI BATTUTO": END
21   IF M>N; IF ABS(M-93)≤2; PRINT "HO QUASI VINTO";:
     GOSUB 30
22   GOTO 4
30   FOR K=1 TO 300: NEXT K: PRINT: RETURN
```

Il segno ■ nella riga 4 è fatto con MODE . SHIFT Z

(restano 2 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
B$	cifra scelta da te
M	punteggio del Casio
N	il tuo punteggio
Z$	cifra scelta dal Casio
$	contiene le scritte di stampa

Dobbiamo confessare di avere avuto qualche remora nello scrivere questo paragrafo, con vari e non ancora disciolti dubbi sull'opportunità o meno di svelare la tecnica del Casio nel scegliere di volta in volta il numero da giocare.

Questo non certo per mantenere nascosto qualche "segreto" del mestiere o presunto tale ma soltanto, enfatizzando un poco, per una questione "morale".

È certamente molto comune, giocando contro un programma non sapendo nulla sul suo funzionamento, attribuirgli poteri (lettura del pensiero, intelligenza,...) che in effetti esso non possiede, e il piacere che si prova spesso nel lasciar libera la fantasia e l'immaginazione può così venir frustrato nel leggere la descrizione delle routine usate e delle operazioni svolte.

Essendo questo volume improntato non soltanto ad intenti giocosi, ma anche didattici, abbiamo optato per la descrizione del modo di operare del programma, in effetti molto semplice, sperando che la disillusione che ne potrebbe derivare possa essere in qualche modo compensata dal piacere di apprendere cose nuove.

Anche se, come specificato nella presentazione, non esiste una strategia vincente, viene però messa in pratica una strategia ottimale: vediamo di che cosa si tratta.

Una branchia della matematica che in questi ultimi tempi ha assunto una sempre maggiore importanza e popolarità è la cosiddetta Teoria dei Giochi. Non lasciarti troppo ingannare dal nome: spesso si tratta di calcoli ed argomenti per nulla ludici, ma qui sufficientemente semplici da permettere la realizzazione del programma. In casi come questo la prima operazione da fare è costruire una tabella, a mo' di tavola pitagorica, che tenga conto di tutte le possibili combinazioni di ogni manche.

La tabella indicherà quanto un giocatore vince o perde rispetto all'altro a seconda dell'esito della prova, e se indichiamo con segno positivo le tue vincite e con segno negativo le vittorie del computer si avrà:

TU

		1	2	3	4	5
C	1	0	—3	2	3	4
A	2	3	0	—5	2	3
S	3	—2	5	0	—7	2
I	4	—3	—2	7	0	—9
O	5	—4	—3	—2	9	0

dove ad esempio il 5 all'incrocio fra la terza riga e la seconda colonna indica che tu acquisti 5 punti se tiri il 2 mentre il Casio controbatte con un 3.

Poichè il gioco è completamente simmetrico per entrambi non può esistere una strategia che consenta di prevalere con certezza; ve ne è soltanto una ottimale che permette, a lungo termine, di avere il pareggio garantito. Senza scendere troppo in dettaglio, sappi che dalla matrice sopra riportata si ottiene un sistema di cinque equazioni in cinque incognite

$$
\begin{array}{rcrcrcrcrl}
0xA & - & 3xB & + & 2xC & + & 3xD & + & 4xE & = 0 \\
3xA & + & 0xB & - & 5xC & + & 2xD & + & 3xE & = 0 \\
-2xA & + & 5xB & + & 0xC & - & 7xD & + & 2xC & = 0 \\
-3xA & - & 2xB & + & 7xC & + & 0xD & - & 9xE & = 0 \\
-4xA & - & 3xB & - & 2xC & + & 9xD & + & 0xE & = 0
\end{array}
$$

il quale sistema, risolto, fornisce cinque pesi (A,B,C,D,E) per le varie manche, cioè in pratica la frequenza con la quale si dovrebbe giocare l'uno, il due, ecc. Il risultato ottenuto è rispettivamente: 10, 26, 13, 16, 1 e quindi su 66 volte (10+26+13+16+1) un buon giocatore dovrebbe, in media, tirare soltanto una volta il numero 5.

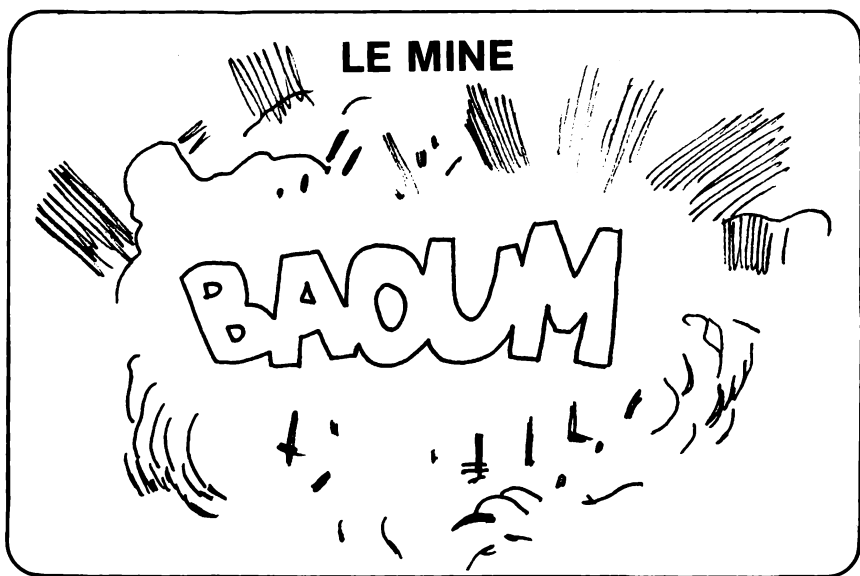
Bene, dopo questa spiegazione dovrebbe esserti chiara la lettura e la comprensione del listato, anche se noi abbiamo variato leggermente i risultati, assegnando al Casio una maggiore probabilità di giocare il 5 poichè la cosa, a nostro parere, rende la gara più divertente. A questo riguardo nella linea 4 la variabile X può assumere un valore massimo di 68 (al posto del 66 "teorico"), ed è su questa base che nelle linee 8/11 si determina la cifra scelta dal computer.

Così, osservando ad esempio le linee 10 e 11 vedrai che Z$ (cifra che verrà messa in gioco) assumerà il valore "4" soltanto se X è compreso fra 49 e 66 (estremi esclusi), il che corrisponde ad una probabilità di 16/68, coerentemente con quanto visto in precedenza.

Il calcolo del punteggio (linee 13/16) non presenta particolarità mentre, sempre nella riga 16, si sfruttano le maggiori possibilità di $, caricato inizialmente, per far variare le esclamazioni e gli improperi che ti sei meritato per esserti fatto "tagliare" così inopinatamente. La variabile J (da 0 a 3) seleziona una delle 4 parole poste in $, che viene poi estratta con la funzione MID. Naturalmente, per applicare questa tecnica, è necessario che tutte le parole abbiano la medesima lunghezza (in questo caso di 6 lettere); se così non fosse occorrerà renderle tali con l'aggiunta di opportuni spazi.

Per concludere la nostra analisi puoi osservare come la linea 21 faccia comparire la scritta "HO QUASI VINTO" soltanto nel caso di un punteggio compreso fra 91 e 95 e, ovviamente, in situazione di vantaggio da parte del computer.

Bene! Ti stai fregando le mani soddisfatto pensando: "ora che so il trucco

lo frego io questo dannato!"? E invece no! Non illuderti troppo, perchè anche adesso sarà molto difficile che tu la spunti. Infatti mentre il tuo Casio è in grado di mettere perfettamente in pratica la teoria, tirando un ipotetico dado a 68 facce, tu non riuscirai mai ad imitarlo, e i numeri da te giocati non potranno rispettare la condizione di casualità pesata richiesta, costringendoti spesso al ruolo di vittima predestinata.

Che lavoro vorresti fare nella vita? Uno tranquillo e privo di responsabilità come dattilografo o impiegato oppure ne preferisci uno più denso di soddisfazioni e, perchè no, di avventura?

Forse propendi per la seconda alternativa, e hai già pensato a qualcosa del tipo pilota d'aereo o corrispondente di guerra, ma prima di prendere decisioni avventate è meglio rifletterci sopra. Se ritieni di amare il brivido, avere nervi saldi, essere un po' pazzo e, nella giusta misura, sprezzante del pericolo, eccoti quindi la possibilità di provare a te stesso se ciò che pensi corrisponde al vero. Il mestiere che abbiamo scelto per te è quello di .....artificiere.

....COME SI GIOCA

Lo scenario è quello di una brulla distesa di terra fangosa in prossimità delle linee nemiche. Tu e i tuoi compagni di squadriglia siete immobili, trattenete quasi il respiro, incuranti delle ultime gocce che ancora cadono dopo una abbondante pioggia.

Siete incappati in un campo minato. Il comandante, senza proferire parola,

ti lancia uno sguardo d'intesa, ma tu già lo sapevi: sarà tutto tuo il compito di togliere le castagne dal fuoco, ed anche velocemente se non vuoi fare una brutta fine.

In tutti gli spazi del display all'inizio compariranno delle cifre, comprese fra il 5 ed il 9, che decrescono rapidamente.

Esse simbolizzano dodici mine in procinto di esplodere, e tu dovrai cercare di disinnescarle prima che ciò avvenga, cioè prima che anche una sola cifra raggiunga lo zero.

Per neutralizzare una determinata mina dovrai posizionarti sopra di essa tramite il cursore (-), che al via della prova appare al centro del display, facendo uso dei tasti "N" e "Z"

N → DESTRA
Z → SINISTRA

Non appena raggiunta la posizione desiderata, solleva subito il dito dal tasto che stai premendo. Solo così la bomba sarà finalmente resa innocua (per il momento) e potrai dirigerti velocemente verso il nuovo obiettivo.

Spostandoti noterai subito che lo spazio sul quale ti eri soffermato non contiene più la cifra precedente, ma un "9". Se ciò non dovesse avvenire significa che la mina non è stata disinnescata e che devi ripetere l'operazione; stai particolarmente attento a quelle poste alle estremità del display perchè sono le più difficili.

La comparsa del "9" significa, forse lo hai già capito, che la mina non è definitivamente disinserita, ma il suo scoppio è solamente ritardato. Quindi dopo aver provveduto a quelle che stanno per esplodere (le cifre più basse) dovrai ritornarvi sopra e così via, in un crescendo sempre più vorticoso di abilità manuale, parallelamente alla sempre maggiore velocità di decremento delle cifre.

Eh! si, dispiace darti una brutta notizia ma questo gioco non ti consentirà, come d'altra parte tutti i videogame in commercio, di uscire vincitore.

Prima o poi, forse anche soltanto per stanchezza (se sarai bravo a raggiungere la massima difficoltà) sarai costretto a cedere disintegrandoti in mille particelle allo scoppio della mina.

Lo scopo del gioco è perciò quello di resistere il più a lungo possibile, e siamo sicuri che il tuo innato spirito di sopravvivenza ti condurrà a ottime prestazioni.

Naturalmente dopo lo scoppio conclusivo, premendo EXE, potrai soddisfare la tua curiosità di conoscere quanto tempo sei "vissuto" ed eventualmente ritentare con una nuova missione.

IL LISTATO
(DEFM 5)

```
  5   PRINT " ƀ ƀLE MINE"
 10   VAC
 20   $="0123456789": P=5
 30   FOR M=0 TO 9
 40   A$(M)=MID(M+1,1)
 50   NEXT M
 60   FOR M=0 TO 11
 70   T(M)=INT(RAN#*5)+5
 80   PRINT CSR M; A$(T(M));
 90   NEXT M: PRINT CSR P; "-";
100    FOR R=1 TO 2 STEP 0
105    N=.5+S: IF N>2; N=2: M=RAN#*12: T(M)=T(M)-1:
      PRINT CSR M; A$(T(M));
110   FOR M=0 TO 11
120   IF KEY≠"N" THEN 130
125   IF P<11; PRINT CSR P; A$(T(P)); CSR P+1; "-";: P=P+1
130   IF KEY≠"Z" THEN 140
135   IF P>0; PRINT CSR P; A$(T(P)); CSR P-1; "-";: P=P-1
140   IF KEY=""; T(P)=9: O=O+1
150   T(M)=T(M)-N: IF T(M)<1 THEN 200
160   PRINT CSR M; A$(INT(T(M))); CSR P; "-";
170   NEXT M
180   S=S+.05
190   NEXT R
200   PRINT: PRINT CSR 3; "BOOM!!"
210   PRINT: PRINT "TEMPO:"; O: GOTO 10
```

(restano 77 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A$(0)...A$(9)	conversione in carattere delle 10 cifre
N	valore per cui verrà decrementata ogni mina
O	tempo trascorso (numero di mine disinnescate)
P	spazio di stampa del cursore
S	aumenta ad ogni ciclo il valore di N
T(0)....T(11)	cifre indicanti le dodici mine

Forse a questo punto del libro, se hai letto tutti i precedenti paragrafi relativi alle spiegazioni dei listati, alcune nostre delucidazioni ti sembreranno superflue e fin troppo facili.

Bene, se ciò si verifica è segno evidente dei tuoi progressi in materia. Noi d'altra parte non possiamo sapere il livello da te raggiunto e quindi proseguiremo come al solito, cercando di chiarire l'algoritmo che dà vita al gioco. Se qualcosa ti risultasse oscura è molto probabile che essa sia stata trattata in precedenti capitoli o nelle appendici e prima di proseguire sarebbe bene che tu cercassi di rintracciarla: senz'altro ti sarà utile per una più proficua lettura.

Ci occuperemo qui di due aspetti del programma: uno relativo alla progettazione vera e propria e che si ripeterà anche in giochi successivi, ed un altro sorto al termine della stesura, dopo le numerose verifiche a cui questo programma (come gli altri del resto) è stato sottoposto prima di considerarlo concluso.

Ti sarà facile comprendere il perchè le cifre, rappresentanti il tempo che resta alle diverse mine prima di esplodere, vadano non soltanto stampate ma necessariamente memorizzate.

Infatti a causa della presenza del cursore (-), ad ogni suo movimento verrà cancellata la cifra a cui esso si sovrappone, e per farla ricomparire è necessario un cosiddetto "rinfresco" del display, che stampi quanto era in precedenza nello spazio in questione.

Per questo i dodici valori scelti casualmente nella linea 70 vengono salvati nel vettore T(M), anche se per ottenerne la stampa in tutti gli spazi verranno usate le corrispondenti conversioni in variabili di carattere A$(T(M)).

Da notare che le relative estrazioni da $, al contrario di come è avvenuto ad esempio per WARGAME, sono state eseguite una volta per tutte nelle prime linee del listato (20/50). In tal modo il gioco risulta senz'altro più veloce, in quanto anche se le cifre cambiano non è più necessario effettuarne nuovamente la conversione (linee 100/190). La dicitura con doppie parentesi A$(T(M)) è stata usata per poter eseguire le stampe all'interno di un ciclo: essendo

A$(0) = "0"
A$(1) = "1"
....
....
A$(5) = "5"
....
A$(9) = "9"

è allora evidente che se ad esempio T(8) = 7, A$(T(8)) selezionerà appunto il carattere "7".

Non ci soffermeremo nè sulla possibilità di usare T(8), fornita dal comando iniziale DEFM 5 che rende disponibili Z(1)...Z(5), nè sul ciclo "infinito" della linea 100 (FOR R=1 TO 2 STEP 0); per maggiori chiarimenti ti rimandiamo alle appendici.

Una piccola notazione riguarda invece l'accennato "rinfresco" del display nelle linee 125 e 135, relative agli spostamenti del cursore nelle due direzioni.

Supponiamo ad esempio che tu intenda muoverlo verso destra (linea 125). Tenendo presente che esso è posizionato nello spazio P, l'operazione sarà possibile soltanto se P stesso è minore di 11 ed in tal caso verrà dapprima stampata la cifra relativa allo spazio in questione (A$(T(P))) e subito dopo verrà aggiornata la posizione del cursore.

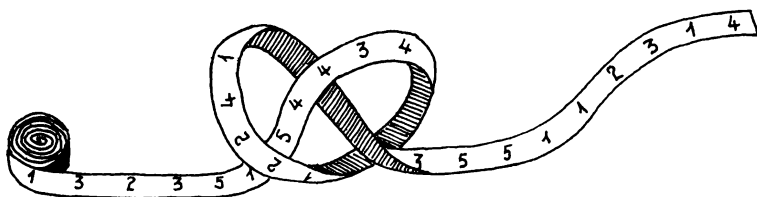
Analoghe saranno le operazioni per quanto riguarda la direzione opposta, mentre nel caso la mina venga disinnescata rilasciando il tasto (nella linea 140), la cifra dello spazio in oggetto viene posta uguale a 9 ed è incrementato il contatore O delle bombe neutralizzate, che stabilisce in un certo senso il tempo di sopravvivenza.

Subito dopo vi sono le istruzioni che "bruciano le micce delle mine". L'ipotesi originaria di decrementare le cifre di una unità alla volta si è dimostrata alla prova dei fatti insufficiente. Anche un giocatore mediocre, adottando una tecnica che non ti diciamo, sarebbe riuscito a resistere praticamente un tempo infinito. Abbiamo quindi dovuto operare diversi cambia-

menti, ed ora ciascuna delle cifre visualizzate viene diminuita di una quantità N, sempre maggiore fino ad un massimo di 2 unità (vedi linea 105). La variabile S che incrementa N viene man mano aumentata nella linea 180; ciò determina un rapido aumento della velocità dopo le prime mosse.

Inoltre quando viene raggiunto il valore limite stabilito, ogni volta che tutte le mine sono state decrementate ne viene selezionata casualmente una (M nella linea 105) e il valore che la rappresenta subirà una ulteriore sottrazione di 1.

In questo modo ci pare proprio che soltanto i più abili potranno veramente resistere a lungo.

SIMON

Pare, secondo autorevoli studi, che con un opportuno allenamento la nostra memoria sia in grado di consentirci prestazioni a dir poco eccezionali. Se vuoi verificare qual è il tuo stato attuale e se hai bisogno o meno di un po' di fosforo, questo è il gioco adatto, che ti consente una ottima ginnastica al cervello.

Vedrai che in poco tempo anche tu sarai in grado di ricordare una sequenza di cifre casuali sulla cui lunghezza massima non avresti scommesso un soldo.

....COME SI GIOCA

Già dal nome avrai certo compreso che si tratta della conversione di un gioco molto in voga, se la memoria non ci inganna, due o tre anni fa. Si trattava di un disco diviso in quattro settori colorati che, a caso, si accendevano producendo un suono, formando così una sequenza via via sempre più lunga che si doveva ricordare e ripetere.

La mancanza nel tuo Casio dei colori e del suono non toglie interesse al divertente passatempo, ed in questo caso sarai chiamato a memorizzare un

numero sempre maggiore di cifre, scelte fra l'1 ed il 5, fino all'inevitabile errore.

Il gioco presenta anche una simpatica particolarità: la classifica dei cinque migliori punteggi e per questo, prima di iniziare il tuo tentativo, ti verrà chiesto il nome.

Verrà verificata la tua eventuale presenza in classifica e in caso positivo si visualizzerà il record personale ottenuto precedentemente.

È sottinteso che, nel caso tu non possegga l'interfaccia per il registratore la sua utilità è molto limitata, essendo valida soltanto per una seduta di gioco. Se, al contrario, puoi utilizzare la comodità della cassetta, sarà sufficiente salvare, prima del programma, i relativi dati (O$...S$,T...X) per riaverli disponibili la volta successiva.

Dopo questa preliminare operazione, premendo EXE, si darà il via alla prova. Cerca di concentrarti a puntino, possibilmente estraniandoti dalla realtà che ti circonda: solo così sarai in grado di verificare le tue capacità e di ottenere un ottimo risultato. Soprattutto non farti distrarre dal fatto che, dopo ogni cifra, è necessario premere EXE prima di impostare quella successiva. Dopo un po' di pratica non è difficile arrivare intorno alle venti, venticinque cifre memorizzate; i più bravi raggiungeranno il massimo di 30 (noi ce l'abbiamo fatta, ed anche tu vedi di non essere da meno).

Varie sono le possibili tecniche per dare una mano alla nostra "materia grigia" qui messa a dura prova, ed ognuno potrà sbizzarrirsi a seconda delle proprie caratteristiche e preferenze.

Per quanto ci riguarda possiamo dire che Anna, per limitare in qualche modo l'"astrattezza" dei numeri, associa ad ognuno di essi la relativa posizione occupata "fisicamente" sulla tastiera, mentre Sandro riesce bene raggruppando l'intera sequenza in gruppi di cinque cifre alla volta.

Come sempre avviene anche qui un minimo di fortuna non guasterà, poichè è indubbio che alcune serie siano più difficilmente memorizzabili di altre.

Al termine della prova, se non hai raggiunto il massimo di 30 caratteri, ti verrà indicato a quale mossa hai commesso l'errore e, premendo EXE, otterrai la stampa della classifica, aggiornata con un eventuale tuo inserimento.

Una doverosa raccomandazione. Non fare come quei videogiocatori da strapazzo che, per veder comparire il loro nome in testa alla classifica, spengono il videogame per cancellare i punteggi precedenti.

Se qualche tuo amico ha fatto meglio di te non fingere un malfunzionamento del nastro per giustificare la perdita dei dati.

Cerca piuttosto di allenarti a dovere e di raggiungere o superare la sua quota "lealmente": certo è più faticoso, ma ti darà anche una ben maggiore soddisfazione.

IL LISTATO

```
  5   A$="1": B$="2": C$="3": D$="4": E$="5"
 10   F=0: INPUT "IL TUO NOME", $
 15   IF LEN($)<7; N$=$: GOTO 25
 20   N$=MID(1,7)
 25   FOR I=1 TO 5
 30   IF N$=N$(I); PRINT "IL TUO RECORD E'"; S(I);: GOTO 50
 40   NEXT I: F=1
 50   PRINT "■ bPREMI EXE b": $=""
 70   G=INT(5*RAN#)
 80   Y$=A$(G)
130   $=$+Y$: L=LEN($)
140   FOR K=1 TO L: J$=MID(K,1)+" b b b"
150   PRINT J$;: NEXT K: PRINT " b bRIPETI b";: FOR K=1 TO L:
      INPUT Z$
190   IF Z$≠MID(K,1); PRINT "ERRORE ALLA MOSSA"; L; " b b b b":
      GOTO 230
210   NEXT K: IF L<30; PRINT "PROSEGUIAMO b b b b";: GOTO 70
215   L=31: PRINT "BRAVISSIMO!!"
230   M=T: H=1: FOR K=2 TO 5
240   IF S(K)<M; M=S(K): H=K
250   NEXT K: IF F≠1 THEN 270
260   IF L>M+1; N$(H)=N$: S(H)=L-1
265   GOTO 275
270   IF L>S(I)+1; S(I)=L-1
275   PRINT "classifica"
280   PRINT: FOR C=1 TO 5
290   IF N$(C)≠""; PRINT N$(C); S(C)
295   NEXT C: GOTO 5
```

Il segno ■ nella riga 50 è fatto con MODE . SHIFT Z
La scritta "classifica" nella riga 275 è fatta in minuscolo (MODE .)

(restano 5 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A$(1)...A$(5)	singola cifra che verrà estratta
H	posizione nel vettore del punteggio minimo
L	mossa corrente
M	punteggio minimo
N$	il tuo nome
N$(1)...N$(5)	i 5 migliori giocatori
S(1)...S(5)	i 5 migliori punteggi
$	sequenza da ripetere

Il gioco in se stesso non presenta particolari difficoltà nè aspetti caratteristici, eccettuato forse il fatto che, essendo la variabile $ utilizzata per memorizzare la sequenza da ripetere, ogni nuova cifra viene scelta in modo diverso da come, forse, ti saresti aspettato. Essa non viene estratta da $, come visto più volte in precedenti giochi, ma è stampato l'elemento del vettore A$() selezionato casualmente nella linea 70.

Puoi inoltre notare come $, prima di essere usato come accumulatore, si renda utile anche in fase di inserimento del nome (linee 10/20). Lo scopo evidente è quello di evitare che l'immissione di un nome troppo lungo causi l'arresto del programma: in una tale eventualità verranno considerati soltanto i primi 7 caratteri.

Qualche parola in più merita invece la gestione della classifica.

Vediamo come è stata organizzata, facendo subito una precisazione: non è stato possibile effettuare un ordinamento, ma i dati vengono visualizzati prima o dopo a seconda del momento in cui sono stati inseriti.

I cinque migliori punteggi vengono memorizzati nelle variabili dalla T alla X, mentre i nomi di chi li ha realizzati occuperanno le stringhe da O$ a S$ costituendo a tutti gli effetti due vettori, S() e N$().

O$ = N$(1) = 1° nome
P$ = N$(2) = 2° nome
Q$ = N$(3) = 3° nome
R$ = N$(4) = 4° nome
S$ = N$(5) = 5° nome

T = S(1) = record di N$(1)
U = S(2) = record di N$(2)
V = S(3) = record di N$(3)
W = S(4) = record di N$(4)
X = S(5) = record di N$(5)

Così al computer (linee 20/30) sarà sufficiente scandire N$() per essere a conoscenza se chi effettua la prova è già presente nella graduatoria o meno e, nel caso, visualizzare il relativo punteggio. È necessario distinguere i due casi (F= 0 o F= 1) poichè ciò dà luogo a diversità di calcolo per quanto riguarda la variazione della classifica e a questo scopo nelle linee 220/250 si ricerca il punteggio minimo presente (M) ed il posto (H) che occupa nel vettore.

Vediamo una volta per tutte come è stata realizzata questa breve routine, che ti sarà senz'altro utile in innumerevoli situazioni: abbiamo semplicemente assegnato a M e ad H i valori relativi al primo elemento (M= T e H= 1), confrontandolo poi con tutti i successivi.

Dopo questa veloce operazione, se ancora non facevi parte della "elite" (F= 1) il numero delle cifre da te correttamente ricordate (L-l) viene paragonato con M. Se hai superato il punteggio minimo (linea 260) sarai di diritto uno dei "magnifici 5" scalzando chi si trovava all'ultimo posto, altrimenti si passerà direttamente alla visualizzazione dei nomi e dei punteggi.

Se i tuoi dati erano già memorizzati (F≠1) viene semplicemente controllato (linea 270) se hai migliorato il tuo record e, nel caso, esso subirà il necessario aggiornamento.

SECRET AGENT 007

Eccoti questa volta nei panni di James Bond, il famoso 007 in missione speciale. Sei infatti all'inseguimento di venti pericolose spie internazionali che cercano di sfuggirti.

Ma la tua abilità nel guidare la formidabile automobile e la mira infallibile ti permetteranno certamente di raggiungerle e colpirle.

Ma attento, se sbagli potresti, una volta tanto, perdere la tua ....partita.

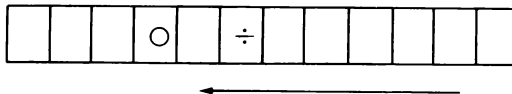
....COME SI GIOCA

Il gioco consiste nell'inseguire e centrare 20 automobili che partono a caso, una alla volta, da una delle estremità del display.

Per muovere quella in dotazione all'agente segreto, rappresentata con "O", hai a disposizione due tasti: T per andare a destra e Q per dirigerti verso sinistra.

Inoltre il tasto E ti consente di sparare ed abbattere le auto nemiche. Queste sono di due tipi:

≠	sono lente e, se colpite, danno 10 punti
÷	sono veloci e valgono 15 punti

Naturalmente devi sparare quando l'unità delle spie si trova nello stesso spazio in cui sei tu; perciò ti consigliamo, appena il nemico compare, di partire senza esitazione, di superarlo, aspettarlo e quindi fare fuoco.

Se lo sparo è a vuoto compare la scritta "CLIC"; dai, cerca di abbattere almeno l'auto successiva che già sta partendo velocemente. Attento anche a fermarti in tempo -al massimo nel penultimo spazio - perchè una prolungata accelerazione ti può mandare fuori pista; in ogni caso ti sono concessi soltanto tre errori e al quarto apparirà la "tragica" scritta NO GOOD.

Il gioco non è considerato buono nemmeno per chi, pur avendolo completato, ha un punteggio inferiore a 240.

Al termine, premendo EXE, oltre al punteggio saprai anche il numero e la qualità delle auto abbattute sotto questa forma

≠	1	4	÷	5			P	:	2	4	0

Per raggiungere il "fatidico" 240 certo dovrai allenarti un po' ed inoltre tener presente alcuni avvertimenti. È importante innanzitutto una partenza immediata se vuoi colpire i nemici prima che si allontanino troppo e scompaiano dal pannello di lettura; perderesti punti preziosi, soprattutto se l'auto è del tipo veloce. Il punteggio è anche in relazione al tempo impiegato ad abbattere l'automobile delle spie, calcolato in base allo spazio in cui essa si trova quando viene colpita. Quindi dovrai cercare di fare fuoco quando è sufficientemente prossima al punto di partenza.

Molta attenzione va prestata all'inseguimento delle auto veloci e, trovandosi persi, conviene aspettarle alla fine della loro corsa, anche se in questo modo ti verrà assegnato un punteggio minore. Puoi sempre rifarti con quelle più lente.

Bene 007, ora puoi partire !!

IL LISTATO

```
 1   PRINT " ƀƀƀSA 007"
 5   VAC
10   IF A=20 THEN 220
15   IF G>3 THEN 250
20   B=RAN#: M=RAN#
30   IF B≤.3; $=" ÷ ": Y=1: GOTO 50
```

```
40    $="+": Y=10
50    C=0: D=11: E=1: F$="○"
60    IF M<.5; C=11: D=0: E=-1
70    FOR I=C TO D STEP E
80    PRINT : PRINT CSR I; $; CSR C; F$;
90    IF KEY="Q"; C=C-2
100   IF KEY="T"; C=C+2
110   IF KEY="E" THEN 150
120   IF C≥0; IF C≤11 THEN 140
130   PRINT: PRINT "FUORI PISTA";: G=G+1: Y=50: GOSUB 300:A
      =A+1: GOTO 10
140   GOSUB 300: NEXT I: A=A+1: GOTO 10
150   PRINT: IF C≠I; PRINT "CLIC";: G=G+1: GOSUB 300: GOTO 210
160   PRINT "BANG-BANG";: GOSUB 300
170   IF $="÷"; P=15: Z=Z+1: GOTO 200
180   P=10: W=W+1
200   J=J+P+(ABS(D-I)-1)↑2
210   GOSUB 300: A=A+1: GOTO 10
220   PRINT: PRINT "ƀƀƀƀFINE"
230   PRINT W; CSR 0; "+"; CSR 3; Z; CSR 3; "÷"; CSR 8; J; CSR 7; "P:"
240   Y=400: GOSUB 300: H$="": IF J≥240 THEN 260
250   H$="ƀNOƀ"
260   PRINT: PRINT H$; "GOOD": GOTO 5
300   FOR N=1 TO Y: NEXT N: RETURN
```

Il segno ÷ nelle linee 30 e 170 è fatto con MODE . SHIFT G
Il segno ○ nella linea 50 è fatto con MODE . SHIFT A
(restano 0 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI

A	contatore delle automobili già apparse
B	numero casuale per la scelta del tipo di auto
C	spazio in cui si trova l'auto di 007
G	contatore degli errori commessi
I	spazio in cui si trova l'auto nemica
M	numero casuale per la scelta del lato di partenza
Y	indice massimo del ciclo su N per la perdita di tempo

Il listato, sebbene relativamente comprensibile, si presta ad alcune considerazioni. Interessante è ad esempio conoscere come si ottenga ogni volta la scelta tra i due tipi di automobile e il lato da cui esse partono.

Automobile e verso di percorrenza vengono determinati tramite la funzione RAN#, come si può osservare nella linea 20.

Tale importante funzione, ricordiamo, fa sì che il computer generi un numero casuale compreso fra 0 ed 1. Osserviamo il listato: se il valore di B (il primo numero casuale) risulta minore di .3, viene presa in considerazione, assegnandola a $ e quindi stampata, l'auto veloce; altrimenti quella più lenta.

Il valore di .3 ha permesso ovviamente di avere una percentuale maggiore di auto lente (70%). Allo stesso modo si è operato per determinare da quale estremità del display l'automobile debba partire.Il computer genera un secondo numero casuale M. Se il suo valore è minore di .5 la vettura partirà alla destra del pannello e si muoverà verso sinistra; se il numero scelto è maggiore di .5 l'automobile avrà senso contrario.

Avendo optato per 0.5 è evidente che in questo caso le probabilità di partenza da destra o sinistra sono entrambe del 50%.

Può essere utile notare come la simulazione della corsa, ottenuta tramite la visualizzazione delle auto nelle diverse posizioni, avvenga comunque con un unico ciclo (linee 50/60/70).

Questo è stato realizzato utilizzando come indici, al posto delle usuali costanti numeriche, le variabili C,D,E

```
FOR I= C TO D STEP E
```

Così, nel caso che l'auto proceda verso sinistra, gli indici minimo e massimo del ciclo risulteranno invertiti (C= 11 e D= 0) rispetto a quelli necessari a far scorrere la macchina da sinistra verso destra (C= 0 , D= 11).

La conseguente differenziazione tra ciclo crescente o decrescente con passo di + o -1 si è ottenuta con l'uso dell'ulteriore variabile E.

Soffermiamoci ora su un altro aspetto del programma.

Per poter inseguire e raggiungere i nemici avvantaggiati dalla anticipata partenza era necessario che l'auto dell'agente segreto avesse una maggiore velocità. Ecco quindi che ogni qualvolta premi uno dei tasti (T o Q)che permettono il movimento della tua auto, la variabile C, che rappresenta lo spazio in cui ti trovi, aumenta o diminuisce (a seconda della direzione presa) di ben due valori.

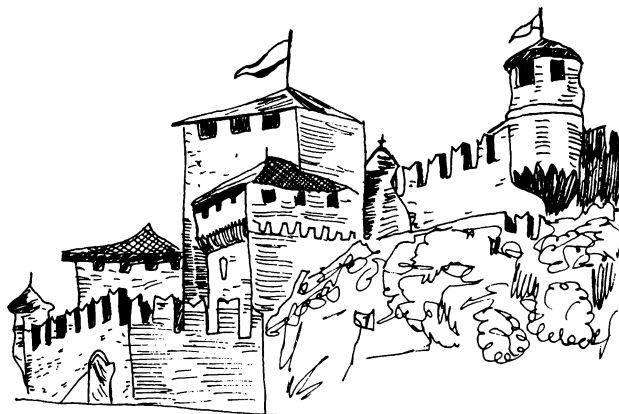
Questo mentre naturalmente i nemici avanzano di una sola posizione alla volta (vedi linee 90,100). Ciò non consente, ed è un fattore da considerare costantemente, di utilizzare l'ultimo spazio a destra o sinistra del display.

Osserviamo ora la linea 200: vi troviamo operazioni di non immediata comprensione che servono comunque per il calcolo del punteggio. Questo infatti dipende, oltre che dal tipo di auto (10 o 15 punti), dalla posizione in cui essa si trova quando viene colpita. Per cui al punteggio di base si aggiungeranno tanti più punti quanto più il nemico è stato abbattuto in prossimità della partenza. La linea 200 permette di assegnare l'abbuono in questo modo:

0,1,4,9,16,25,36,49,64,81 punti.

Nella formula, D - I indica la distanza della macchina dal termine della pista nel momento in cui viene colpita; questo valore viene reso sempre positivo tramite la funzione ABS, e viene decrementato di un'unità poichè l'ultimo spazio del display non è accessibile all'auto di 007.

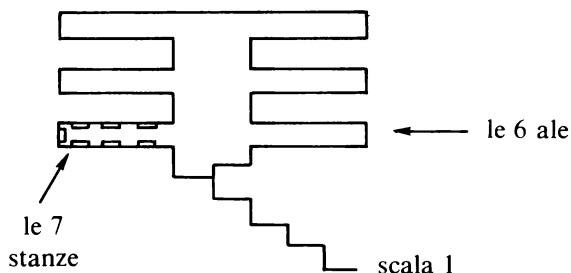
Dopo questo calcolo si ottiene un numero compreso tra 0 e 9, il cui quadrato costituirà l'abbuono.

LE 7 CHIAVI

A questo punto ti aspetta una prova impegnativa e pericolosa, a metà tra il mondo fantastico della fiaba e la realtà storica.

Siamo in pieno Medioevo, epoca oscura e barbara, ma anche ricca di coraggiosi ed indomiti cavalieri che mettono la loro spada al servizio dei deboli.

Ordunque Messere, una bellissima principessa si trova prigioniera in un immenso e misterioso castello, sorvegliata da terribili draghi che uccidono con il loro alito di fuoco. Non vorrai per caso abbandonarla al suo triste destino?! Balza sul tuo cavallo bianco, brandisci la spada e affronta per lei insidie e pericoli.

....COME SI GIOCA

Il castello nel quale si trova prigioniera la principessa è composto da tre grandi scalinate, ognuna delle quali conduce a sei ale del palazzo; su ogni ala si aprono sette stanze.

In totale le stanze sono quindi 126.

Il gioco consiste nell'aggirarsi nel maniero e trovare le sette chiavi necessarie a liberare la fanciulla, il tutto entro il tempo massimo consentito. All'inizio della prova, dopo la visualizzazione del titolo, premendo EXE ci sarà qualche istante di attesa, necessario al computer per la "costruzione" del castello e quindi compariranno in successione le richieste di quale scala (dall'1 al 3), ala (dall'1 al 6) e stanza (dall'1 al 7) si intende esplorare.

Dovrai indicare la tua scelta e dal calcolatore ti verrà il responso: saprai di che tipo è la stanza nella quale sei entrato nella tua affannosa ricerca, mentre il tempo a disposizione continua a calare piuttosto velocemente.

Le possibilità che ti si presentano sono:

— la stanza è chiusa "■"; indicane subito un'altra non appena comparirà la scritta relativa;
— la stanza è vuota "[ ]"; dai, prosegui nell'esplorazione del castello;
— la stanza è occupata da un terribile drago "¥" che devi uccidere; per causare la sua morte premi un tasto qualsiasi il più velocemente possibile perchè ogni attimo di esitazione ti costerà una notevole perdita di preziosi istanti;
— nella stanza scelta hai trovato la chiave: sei stato molto fortunato, ma mettiti subito alla ricerca delle altre.

Tieni presente che non è facile riuscire a totalizzare le sette chiavi necessarie nel tempo consentito e molte volte vedrai comparire sul pannello la scritta "TROPPO TARDI".

I tuoi riflessi saranno messi frequentemente alla prova, poichè risulta determinante la prontezza che avrai nell'uccidere il drago; per la riuscita del gioco è comunque indispensabile una buona dose di fortuna che ti permetterà di indovinare velocemente le stanze nelle quali sono racchiuse le chiavi.

I tuoi tentativi di ricerca potranno seguire un certo ordine logico (ad esempio puoi optare per tutte le stanze della prima scala, prima ala e così via),

oppure puoi procedere casualmente ed a salti, cercando in questo caso di ricordare le scelte già indicate.

Per un gioco più movimentato e meno monotono ti consigliamo la seconda alternativa che tra l'altro, dopo diverse prove, è risultata anche quella più conveniente e con maggiore probabilità di esito favorevole.

IL LISTATO

```
 1    FOR Z=0 TO 17: A$(Z)="0000000": NEXT Z: PRINT
      "LE 7 CHIAVI": U=800
 2    X$="1": Y=40: GOSUB 50: X$="2": GOSUB 50
 3    X$="3": Y=20: GOSUB 50: T=0
 4    IF U<5; PRINT "TROPPO TARDI": END
 5    U=U-5: INPUT "SCALA (1/3)",X: IF X>3 THEN 5
 6    INPUT "ALA (1/6)",Y: IF Y>6 THEN 6
 7    INPUT "STANZA (1/7)",W: IF W>7 THEN 7
 8    S=6*X+Y-7: $=A$(S): V$=MID(W,1): V=9+3*VAL(V$): PRINT:
      GOTO V
 9    PRINT "VUOTA ƀ[]";: GOTO 60
12    PRINT "CHIUSA ƀ■";: GOTO 60
15    PRINT "DRAGO ƀ¥";
16    IF KEY=""; U=U-10: GOTO 16
17    PRINT: PRINT "UCCISO!";: GOTO 60
18    PRINT "CHIAVE!";: T=T+1: X$="0": Y=1: GOSUB 50
19    PRINT CSR 11; " ƀNE HAI"; T;: GOTO 60
50    FOR Z=1 TO Y: IF Y≠1; S=INT(RAN#*18): W=INT(RAN#*7)+1
51    $=" ƀ"+A$(S)+" ƀ": $=MID(1,W)+X$+MID(W+2)
52    A$(S)=MID(2,7): NEXT Z: RETURN
60    FOR Z=1 TO 200: NEXT Z: PRINT: IF T<7 THEN 4
61    PRINT " ƀE SALVI LA PRINCIPESSA"
```

I segni [e] nella riga 9 sono fatti con MODE . SHIFT T e Y
Il segno ■ nella riga 12 è fatto con MODE . SHIFT Z
Il segno ¥ nella riga 15 è fatto con MODE . SHIFT R

(restano 2 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A$(0)...A$(17)	le 18 ale del castello
T	numero delle chiavi trovate
U	tempo a disposizione
X$	determina il tipo di stanza
W	stanza richiesta
X	scala richiesta
Y	dapprima indica quante volte un determinato tipo di stanza viene immesso nel castello ed in seguito rappresenta l'ala richiesta.

Ora che hai trovato le sette chiavi e liberato la principessa, puoi dedicarti con maggior tranquillità alla lettura e comprensione del listato, che non mancherà di fornirti utili indicazioni per progredire nel tuo Basic.

Ti consigliamo di prestare particolare attenzione alle modalità con le quali il castello viene "costruito" dal computer, poichè tecniche simili si ripeteranno anche in programmi successivi.

In effetti il gioco poteva essere concepito senza un vero "disegno" della fortezza, semplicemente assegnando ad ogni tuo tentativo una certa probabilità di trovare la chiave, il drago od altro.

Questa soluzione non ci è sembrata del tutto soddisfacente, mancando una qualsiasi corrispondenza tra la stanza selezionata e la sua effettiva realtà fisica. Anche se in fondo si tratta di un problema principalmente psicologico, abbiamo preferito dar vita alla simulazione di un castello.

Questo consta di ben 126 stanze e la loro memorizzazione non sarebbe stata di certo possibile facendo uso di variabili numeriche. Ogni stanza, d'altra parte, deve essere univocamente individuabile e quindi necessita di una locazione di memoria ad essa riservata.

Abbiamo così sfruttato la capacità data dalle stringhe di contenere 7 caratteri e 18 variabili sono state sufficienti a rappresentare l'intero castello.

Infatti esso, con un po' di fantasia, può essere pensato come una tabella di 18 righe (le 18 ale), nella quale ogni riga contiene 7 elementi (le 7 stanze). Le variabili necessarie sono state utilizzate per praticità sotto forma di componenti del vettore A$(Z) e, a conclusione della linea 1 del listato, si ottiene la seguente situazione:

Come puoi osservare, ogni casella è stata riempita inizialmente con il carattere " 0 " che sta ad indicare "stanza vuota".

In seguito molte di quelle caselle verranno occupate da un cifra diversa, 1, 2 o 3 indicanti rispettivamente: stanza chiusa, drago e la presenza della tanto sospirata chiave.

Questa operazione viene effettuata nella subroutine 50, richiamata per tre volte (linea 2 e 3) e preceduta dai parametri X$ e Y. X$ indicherà la nuova cifra da immettere, mentre Y si riferisce al numero delle volte che essa verrà introdotta.

Un ragazzo "sveglio" come te avrà già osservato che, mentre per il primo richiamo i 40 "1" si sovrapporranno ad altrettanti "0" di partenza, per i successivi capiterà senz'altro che un "2" possa prendere il posto di un "1". In questo modo non si sarà affatto verificata la ripartizione fra chiavi, draghi e stanze chiuse di 20, 40, 40 come potrebbe sembrare ad uno sguardo superficiale.

Analizziamo ora la subroutine 50.

In essa la locazione selezionata casualmente tramite S e W (S determinerà l'ala e W la stanza) subirà appunto una delle modifiche sopra citate.

La variazione del vecchio carattere 0 avviene nelle linee 51 e 52, dove viene attuata la seguente sequenza:

— la linea della tabella selezionata da S viene trasferita in $ al fine di poter applicare la funzione MID.
— $ viene modificata con l'inserimento di X$, lasciando invariati tutti i caratteri precedenti e seguenti la locazione scelta.
— $ così modificata viene di nuovo trasferita in A$(S).

Nota che, nel caso venga selezionato il primo (o l'ultimo) carattere di una linea, si potrebbe incorrere in un errore, causato dal tentativo di estrarre quelli precedenti (o seguenti) che in realtà non esistono. Per ovviare a ciò si è utilizzato il "trucco" di aggiungere temporaneamente due spazi, uno all'inizio ed uno alla fine della variabile $.

Dopo il terzo ritorno dalla subroutine 50 la tabella rappresentante il castello sarà composta da caselle di vario tipo

A$(1) = 0 1 0 3 0 2 1

e sei così pronto ad iniziare il gioco.

Passiamo ora a chiarire la procedura con la quale, partendo dai dati forniti in input, si arriva a stabilire la collocazione nella tabella della stanza che intendi visitare; la formula della linea 7 ha esattamente questo scopo. Per meglio comprenderla tieni presente che essa è una semplificazione di quella originaria data da $S = 6 * (X-1) + Y - 1$

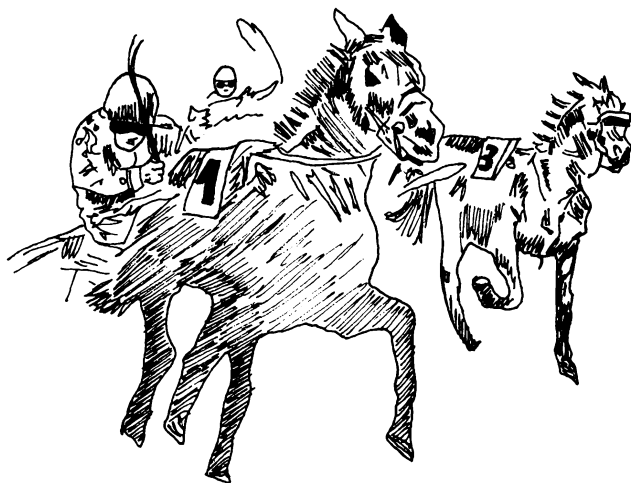
Con la moltiplicazione si individua la scala richiesta e successivamente ci si posiziona nell'ala desiderata, tenendo conto che è necessaria la sottrazione di 1 poichè la numerazione delle ale inizia da 0 ed è sfalsata di 1 rispetto ad Y.

Se, ad esempio, $X = 1$ e $Y = 3$ si ha $S = 6 * (1 - 1) + 3 - 1 = 2$ che individua quindi il terzo elemento del vettore: A$(2); si accederà quindi alla stanza estraendo il corrispondente carattere della variabile sopra selezionata.

Un accorto uso dell'istruzione GOTO nella linea 7 permetterà di ricevere informazioni relative alle stanze, evitando l'uso di successivi IF....THEN.

Il calcolo di V consente, a seconda del carattere trovato nella casella richiesta, di "saltare" alla linea corrispondente: 9 - 12 - 15 - 18.

L'ultimo piccolo problema era quello di togliere la chiave dalla stanza al momento del suo ritrovamento, per evitare una ripetuta richiesta della stessa. Si è semplicemente usufruito della solita subroutine 50, saltando in questo caso alla linea 51, dove il carattere X$ (= "0") verrà a sostituire il precedente "3" indicante la chiave.

DERBY

Tramite vie non troppo lecite hai saputo da un amico book-maker che il fantastico purosangue "ASTERIX", grande favorito per la corsa di domani pomeriggio, non è in perfetta forma.

Hai finalmente una buona occasione per vincere una considerevole sommetta puntando tutto ciò che possiedi su "BACICCIA". È un cavallo di certo meno bravo, quotato 1 a 4, ma che potrebbe anche spuntarla considerata la situazione a lui favorevole. Senz'altro nessuno prevede la sua vittoria ed una buona giocata ti permetterà forse anche di "ripulire" completamente il botteghino.

Ma attento, se rimani invece senza una lira non versare poi lacrime di coccodrillo!

....COME SI GIOCA

E così caro amico ti ritrovi oggi tra bellissimi cavalli che galoppano più veloci del vento, persone che urlano, altre che si agitano ed altre ancora che osservano la corsa con il classico binocolo. Se ti piace questa stimolante

atmosfera dello stadio ippico, vedrai che anche il nostro pur semplicissimo "Derby" non mancherà di appassionarti.

È un gioco per due persone che simula la corsa di tre cavalli, rappresentati sul pannello dalle lettere A, B e C.

Inizialmente tu ed il tuo amico dovrete inserire, in risposta alle domande del computer, i rispettivi nomi facendo bene attenzione a non digitare più di sette lettere, immissione che provocherebbe all'istante il blocco del programma. Potrai quindi conoscere le quotazioni relative ai tre cavalli, visualizzate in questa forma:

Saprai certamente il significato di tali scritte: esse indicano quante volte (in questo caso 3) la tua puntata viene moltiplicata in caso di vincita e ti forniscono utili indicazioni sulla bravura dei cavalli. Infatti quanto più un corridore è veloce e ha quindi maggiori probabilità di arrivare primo, tanto meno la tua scommessa ti verrà accreditata, poichè la vittoria era quasi certa. Ad esempio le quotazioni

A = 1: 4
B = 1: 2
C = 1: 5

ti dicono subito che il cavallo favorito è B.

È quindi chiaro che una puntata su tale purosangue comporta sì una vincita minore (la posta sarà solo duplicata), ma ciò è ampiamente controbilanciato da una maggior sicurezza di risultato positivo e a questo punto sta solo a te decidere i rischi che vuoi correre.

Tieni anche presente che la corsa dei cavalli reca una componente casuale, proprio come in un vero Derby, dove il favorito può essere in una giornata "no" e condurre una pessima gara. Ma torniamo a noi.

Il computer informa il primo giocatore della cifra che ha a disposizione, somma inizialmente posta a L. 100.000, e gli chiede quindi di effettuare la sua giocata attraverso un input, prima del cavallo preferito e poi della quota che intende scommettere; la stessa operazione si ripete per l'avversario.

Ed ecco che entriamo ora nel vivo del gioco: potrete cioè assistere all'avvincente corsa dei cavalli, tifando ognuno per il proprio beniamino.

Dopo la scritta "PARTENZA", premendo EXE, vedrai le tre lettere A, B e C partire da sinistra e scorrere sul display inseguendosi, sovrapponendosi,

distanziandosi di nuovo come in una vera gara, finchè una di esse raggiungerà l'estremità destra, tagliando così il traguardo.

Hai azzeccato il cavallo "giusto" e stai vincendo?

Benissimo, allora tocca a te iniziare la nuova manche premendo EXE; infatti, per evitare plagi di puntate, abbiamo fatto in modo che ad ogni turno iniziasse a scommettere il giocatore che in quel momento possiede più denaro.

Il gioco continua con la somma a disposizione dei due avversari sempre aggiornata attraverso il calcolo delle perdite e delle vincite, fino a quando uno dei partecipanti rimane senza contante e non può quindi effettuare alcuna puntata.

Con questi presupposti vedrai che ti sarà impossibile non divertirti con questo "Derby" insieme al tuo amico e naturalmente al fedelissimo CASIO.

IL LISTATO

```
2    PRINT "DERBY": FOR K=1 TO 5 STEP 4: INPUT "NOME",
     A$(K): B(K)=1E5: NEXT K
4    R$="A": U$="B": X$="C": PRINT "QUOTAZIONI";
5    O=0: FOR K=1 TO 7 STEP 3: R(K)=INT(RAN#*4+2): S(K)=0:
     GOSUB 99
6    PRINT: PRINT Q$(K); "=1:"; R(K);: GOSUB 99: NEXT K
7    L=1: M=5: IF C<G; L=5: M=1
8    FOR K=L TO M STEP M-L: GOSUB 30
12   NEXT K: IF O=1; END
13   PRINT "PARTENZA"
14   FOR K=1 TO 7 STEP 3: S(K)=S(K)+RAN#*(15-R(K))
15   FOR J=12 TO 0 STEP -1: IF S(K)<10*J; NEXT J
16   IF J≠12; PRINT CSR J; Q$(K);: NEXT K: GOSUB 99: PRINT:
     GOTO 14
17   PRINT: PRINT "VINCE:";: GOSUB 99: PRINT Q$(K)
18   FOR P=1 TO 5 STEP 4: IF C$(P)=Q$(K); B(P)=B(P)+D(P)*R(K)
19   IF B(P)=0; O=1
21   NEXT P: IF O=0 THEN 4
22   PRINT "FINE": GOTO 8
30   PRINT: PRINT A$(K); ":"; B(K);: GOSUB 99: IF O=1 THEN 34
31   PRINT: INPUT "CAVALLO", C$(K)
32   INPUT "QUANTO", D(K): IF D(K)>B(K) THEN 32
33   B(K)=B(K)-D(K)
34   RETURN
99   FOR A=1 TO 200: NEXT A: RETURN
```

(resta 1 passo)

SE VUOI IMPARARE

Vedi ora di dedicarti con attenzione all'esame del listato sperando che al termine, stordito dalle nostre spiegazioni, tu non decida di "darti all'ippica".

Già da un primo sguardo al programma ti sarai accorto dei numerosi cicli FOR....NEXT sparsi qua e là e, a parte quello presente nella linea 15 di cui ci occuperemo in seguito e l'iterazione di ritardo della subroutine 99, gli altri possono essere distinti in due tipi.

I cicli che si riferiscono ai giocatori sono costituiti da una variabile indice che, con uno STEP di 4, assume i valori 1 e 5.

Tale passo deriva dalla necessità di riservare, per ognuno dei due partecipanti, quattro variabili che contengano i rispettivi dati e precisamente: nome, somma posseduta, cavallo su cui punta e cifra scommessa

	NOME	SOMMA POSSEDUTA	CAVALLO	SCOMMESSA
1° GIOCATORE	A$(1)	B(1)	C$(1)	D(1)
2° GIOCATORE	A$(5)	B(5)	C$(5)	D(5)

L'unica nota al riguardo è costituita dalle linee 7 ed 8, che risolvono la necessità di far scommettere per primo colui che attualmente è in vantaggio, come ricorderai avendo già letto il "Come si gioca".

In seguito al test della linea 7 il ciclo su K andrà da 1 a 5 con incremento di 4 oppure da 5 ad 1 con passo di -4, a seconda che la somma posseduta dal primo giocatore (C) superi o meno quella a disposizione del secondo (G).

Ciò comporta (nella subroutine 30) l'invito ad iniziare la puntata sulla corsa imminente per A$(1) o A$(5).

L'altra categoria di statement FOR interessa invece ciò che si riferisce ai tre cavalli e consta di un valore iniziale di 1, quello finale di 7 ed uno step pari a 3.

I rispettivi nomi, quotazioni e punteggi che determineranno l'avanzare nella pista sono così organizzati:

CAVALLO	NOME	QUOTAZIONE	PUNTEGGIO
A	Q$(1)	R(1)	S(1)
B	Q$(4)	R(4)	S(4)
C	Q$(7)	R(7)	S(7)

La parte forse meno comprensibile del programma è quella relativa alla corsa vera e propria (linee 14/16).

Ogni cavallo ha un "punteggio" che man mano aumenta durante la gara e che stabilisce l'avanzare nel display della lettera che lo rappresenta. Questo valore S(K) è determinato in parte dalla casualità ed in parte dalla "forza" del purosangue e così, ad esempio, al primo "passaggio" potrà risultare al massimo 12,9999... per chi è quotato 1: 2 (R(K)=2) mentre 9,9999.... se la quotazione è di 1: 5 (R(K)=5).

In seguito, tramite il ciclo su J nella linea 15, S(K) è confrontato con successivi e sempre minori multipli di 10, partendo da 120 (10 x 12) e finchè il suo valore rimane inferiore a 10 x J il ciclo continuerà fino ad arrivare eventualmente a 0 (10 x 0). In caso contrario (S(K)≥10 x J) il cavallo interessato verrà visualizzato nello spazio J.

In tal modo, tanto maggiore risulta S(K) e quindi la bravura del "corridore" quanto più questi occuperà una posizione avanzata.

Avrai infine di certo compreso che, per i motivi sopra indicati, benchè il ciclo su J abbia un passo negativo, la corsa in realtà si svolge dalla sinistra alla destra del display e avrà termine quando un cavallo dovrebbe essere stampato in CSR 12.

ZIC-ZAC

È stata una battaglia terribile e cruenta, ormai sei rimasto solo a difendere il castello dall'assedio dei nemici che continuano ad avanzare. Attento, un arciere ha già iniziato la scalata alle alte mura di cinta, per penetrare nella fortezza: se riesce nel suo intento non vedrai di certo spuntare l'alba di domani. Per difenderti hai a disposizione solo pentoloni di olio bollente; non è molto ma forse ti basteranno a far indietreggiare l'avversario, finchè cadrà nel profondo fossato che circonda il castello.

Ma, anche se il primo arciere è stato sconfitto, ecco arrivarne subito degli altri sempre più abili, precisi e coraggiosi.

Forza con i pentoloni, più veloce! Forse prima o poi arriveranno i tanto sospirati rinforzi.

....COME SI GIOCA

Questa nuova avventura, che ti darà la possibilità di "esplorare" i tuoi riflessi e la tua abilità manuale, risulta avvincente ed interessante non solo per il giocatore attivo ma, a nostro avviso, anche per gli spettatori.

Puoi giocarci dovunque: in viaggio, durante le vacanze e naturalmente a casa e presto diventerai talmente in gamba da riuscire ad eliminare tutti gli arcieri che tentano la scalata al castello, risultato inizialmente molto improbabile. Ma stai ora bene attento a come devi procedere.

Sul pannello di lettura si visualizza una situazione di questo tipo:

6					←	Ω					0

Tu ti trovi all'estremità destra, nello spazio inizialmente occupato dal numero 0; c'è un arciere munito di freccia al centro ed ancora un numero compreso tra 0 e 9, scelto casualmente dal computer, nel primo spazio.

Per allontanare il pericolo che incombe e versare l'olio bollente sul nemico devi utilizzare il tasto ".".

Ad ogni pressione il numero di destra viene incrementato di una unità e, quando il suo valore si uguaglia a quello del numero casuale, devi intervenire con la massima tempestività e bloccare il tutto tramite il tasto "Z".

6					←	Ω					6

Se questa operazione ti è riuscita, significa che hai centrato il tuo bersaglio e vedrai l'arciere, raggiunto dall'olio, allontanarsi di uno spazio verso sinistra. Bene, per il momento sei salvo! Far indietreggiare il nemico non è però semplice come sembra perchè dopo un certo tempo, a dire il vero molto breve, il numero visualizzato scompare per lasciare il suo posto ad un altro e se non hai ottenuto prima la necessaria uguaglianza l'arciere si dirigerà inesorabilmente verso di te, avanzando così nella sua scalata. Abbiamo inoltre posto un limite di tempo scaduto il quale, anche se la freccia dell'avversario non ti ha ancora colpito, hai perso comunque.

Avrai invece la vittoria se riesci a ricacciare l'omino all'estremità sinistra del display ed egli sprofonda così nel fossato. Superata una prova, accedi automaticamente a quella successiva; i livelli di difficoltà progressivamente proposti sono nove e ogni volta il tempo a disposizione per ottenere l'equivalenza tra i due numeri diminuisce, perchè le cifre scelte casualmente appaiono e scompaiono ad una aumentata velocità.

Le manche sono così sempre più difficili e ti sono richiesti interventi di volta in volta più rapidi e pronti.

Ti basterà provare il gioco per capire che, tramite ripetute pressioni del tasto ".", anche se velocissime, non riusciresti mai a portare in tempo il numero che ti rappresenta alla stessa quantità di quello casuale, soprattutto ai livelli più alti.

Non ti resta quindi che mantenere schiacciato il tasto in modo che il tuo numero possa raggiungere in pochi istanti il valore desiderato, anche se in questo caso il velocissimo susseguirsi delle cifre renderà molto più complicato bloccare tempestivamente la loro rotazione al momento giusto.

Ora che hai imparato il meccanismo del gioco, diamo una breve occhiata al punteggio. Esso è in relazione inversa al tempo che impiegherai a sconfiggere l'arciere, causando la sua caduta nel fossato e andrà da un massimo di 50 punti, se riesci subito e senza errori, ad un punteggio minimo di 0 nel caso di tempo scaduto, che non ti permetterà ovviamente di passare alla prova successiva. Tutto O.K.? Allora riscalda ben bene l'olio ché il primo arciere è già in arrivo!

IL LISTATO

```
   5  PRINT " ƀƀZIC-ZAC"
  10  VAC
  15  FOR X=1 TO 9
  16  W=60
  20  PRINT: PRINT "LIVELLO"; X;: FOR V=1 TO 300: NEXT V: PRINT
  25  L=50-3*X: D=5
  30  A$="Ω→": B$="←Ω":$="0123456789"
  40  PRINT CSR 10; C; CSR D; B$;
  50  R=INT(RAN#*10): Z$=MID(R+1,1): PRINT CSR 0; Z$;
  60  FOR N=1 TO L: IF KEY="."; GOSUB 500
  70  IF KEY="Z"; IF C=VAL(Z$) THEN 600
  80  NEXT N
  90  D=D+1
 100  IF D=11; PRINT: PRINT "PROVE SUPERATE:"; X-1;
      " ƀƀƀTOTALE=";P: END
 150  GOTO 630
 500  C=C+1: IF C=10; C=0
 510  Y$=MID(C+1,1): PRINT CSR 11; Y$;: RETURN
 600  D=D-1: IF D=0 THEN 1000
 630  Y$=MID(C+1,1): W=W-2.5: IF W<0; D=11: GOTO 100
 650  PRINT: IF D≤5; PRINT CSR 11; Y$; CSR D; B$;: GOTO 50
 700  PRINT CSR 11; Y$; CSR D-1; A$;: GOTO 50
1000  PRINT: PRINT "PUNTI"; W;: FOR V=1 TO 300: NEXT V: P=P+W:
      NEXT X
1100  PRINT: PRINT "BRAVISSIMO!! ƀƀTOTALE="; P: END
```

Il segno Ω nella riga 30 è fatto con MODE . SHIFT N

Il segno → nella riga 30 è fatto con MODE . SHIFT P

Il segno ← nella riga 30 è fatto con MODE . SHIFT I

(restano 53 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
C	cifra da te controllata
D	spazio di stampa dell'arciere
L	velocità del gioco
P	punteggio conclusivo
R	cifra che devi uguagliare
W	punteggio di ogni livello
X	livello di difficoltà

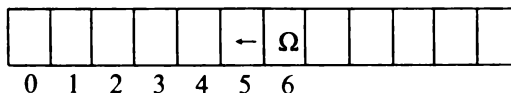
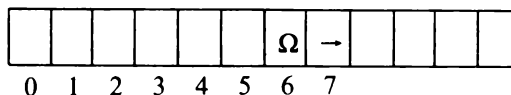
Dopo aver dato libero spazio alla fantasia, tra arcieri e pentoloni di olio bollente, torniamo ora ai problemi di noi "computeromani" e dedichiamoci all'esame del listato.

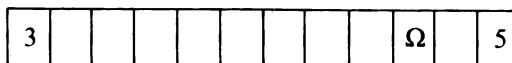
I nove livelli del gioco a difficoltà crescente sono stati realizzati con un ciclo su X da 1 a 9; in ognuno di essi il tempo sempre minore per cui la cifra casuale rimane visualizzata è fornito dalla variabile L (linea 25).

Variando X essa assumerà valori compresi tra 57 (per X=1 nel primo livello) e 23 (per X=9 nell'ultimo livello), determinando una diversa durata del ciclo su N nella linea 60, intervallo che coincide appunto con il tempo di permanenza della cifra sul display.

La moltiplicazione per 3, nella formula per il calcolo di L, ha lo scopo di determinare una più accentuata differenza tra le varie prove, causando ad esempio nell'ultima una velocità doppia rispetto alla prima.

Proseguendo noterai (linea 30) che per rappresentare il tuo nemico arciere sono state usate due variabili, A$ e B$. La prima, contenente la stringa "Ω→", verrà utilizzata nel caso in cui l'arciere è in una posizione di vantaggio quando cioè, avendo superato la metà del display, si trova più vicino alla cima del castello che al fossato; B$ (←Ω) sarà ovviamente richiamata nella situazione contraria. Prima che si visualizzi la nuova posizione del nemico verrà quindi effettuato un controllo (linea 650) relativo allo spazio di stampa D, che seleziona adeguatamente una delle due variabili. Poichè volevamo che in entrambi i casi nello spazio D fosse stampata la freccia, che è quella che interessa, mentre B$ viene direttamente visualizzata in CSR D, per A$ si è dovuto operare con D-1.

Se D=5 appare B$ in CSR 5

Se D=7 appare A$ in CSR 7-1, cioè CSR 6

Anche la stampa dei due numeri che entrano in gioco alle estremità del display ha creato qualche lieve complicazione, risolta nel solito modo che consiste nell'estrarre da $ la cifra corrispondente. Questa necessità risulta abbastanza ovvia per quanto riguarda il numero casuale R, convertito in carattere (Z$) al fine di ottenerne la visualizzazione nel primo spazio del pannello.

Più sottile è la motivazione riferita al numero di destra; anche se apparentemente sembrerebbe inutile convertire C in Y$ (linee 510 e 630) ciò è stato invece indispensabile. Infatti lo spazio vuoto riservato al segno verrebbe a sovrapporsi alla freccia nell'ipotesi in cui l'arciere riesca a conquistare il penultimo spazio accessibile (CSR 9)

la freccia non comparirebbe perchè cancellata dal segno sottinteso +

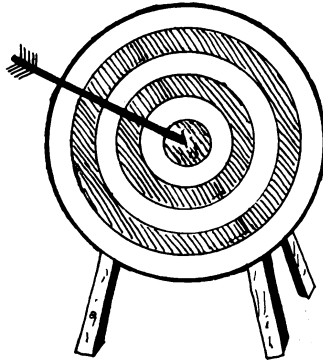
Vediamo infine qualcosa sul punteggio (linea 630).

Il valore di W, posto inizialmente a 60, ad ogni spostamento del nemico subisce una diminuzione di 2,5 indipendentemente dal lato verso cui si sta dirigendo. Ti sarà quindi chiaro che i punti totalizzati sono inversamente proporzionali al numero delle mosse impiegate a respingere l'attacco dell'arciere e mandarlo "fuori" dal display.

Il valore di 2,5 da noi adottato consente di avere sempre un punteggio multiplo di 5; infatti il numero degli spazi che l'omino percorre prima di cadere nel fossato è sempre pari, comunque risulti l'eventuale alternanza di errori e successi.

Una serie numerosa di questi tentativi determinerà il rapido raggiungimento dello 0 e la conseguente fine del gioco, come se il nemico avesse guadagnato la tua postazione (IF W<0; D=11).

Se i tuoi riflessi ti hanno invece consentito di realizzare un'ottima prova esente da errori, il valore iniziale di 60 verrà decrementato solo quattro volte ed otterrai così la bellezza di 50 punti.

BERSAGLI

C'era una volta un abilissimo arciere di nome Guglielmo Tell (questa l'ho già sentita) che, grazie alla sua infallibile mira, riuscì a centrare una piccola mela posta sul capo del figlio, diventando così famoso in tutto il mondo.

Ed ora rispondi: vorresti impersonare il padre, il figlio o la "povera" mela?

Scherzi a parte, questo nostro gioco non ti farà certo diventare bravo come il leggendario eroe svizzero, ma di sicuro la tua mira ne uscirà più precisa ed efficace.

Scaglia ora la prima freccia dritta al centro del bersaglio!

....COME SI GIOCA

Nel gioco che ti presentiamo, tranquillizzati, non ti sarà sicuramente richiesto di tirare ad una mela posta sulla testa di chicchessia, ma semplicemente di centrare dei bersagli scoccando la tua freccia con la maggior precisione possibile.

La situazione iniziale è data dalla stampa sul pannello di lettura di 6 cifre, comprese tra l'1 ed il 9; cifre visualizzate con l'intervallo di uno spazio tra l'una e l'altra.

Talvolta può apparire, in sostituzione di alcuni numeri, il carattere ":" il cui significato potrai tra breve conoscere. In ogni caso i sei simboli visibili rappresentano i bersagli che tu hai il difficile compito di colpire ed abbattere, dopo aver indicato il numero di tiri che intendi effettuare.

8		9		:		3		7		5	

Questo è un esempio di come può apparirti il display; non ti resta quindi che tendere la corda dell'arco e scagliare la freccia sugli obiettivi da colpire.

C'è infatti un rombo "◆" che scorre velocemente e di continuo da una parte all'altra del pannello; in effetti la sua corsa è talmente rapida che, più che vederlo chiaramente, puoi intuirne la posizione dal balenio che lascia al suo passaggio.

Esso rappresenta una specie di mirino che si porta di volta in volta sopra ad ognuno dei bersagli fissi; tu dovrai abbatterli, tramite la pressione di un tasto qualsiasi, proprio quando questi si trovano esattamente al centro del "mirino", cioè quando si avrà una sovrapposizione con la cifra da colpire.

Se l'operazione ti è riuscita otterrai un punteggio pari al numero in questione che verrà subito sostituito da un nuovo valore.

Frequentemente ti capiterà invece di fallire il bersaglio e premere il tasto quando il "◆" si trova in corrispondenza di uno spazio vuoto: in questo caso sarai penalizzato di 3 unità.

Particolare attenzione dovrai inoltre prestare ai ":" cercando di evitarli assolutamente; infatti, se colpiti, determineranno l'annullamento del punteggio guadagnato con tanta fatica.

Un piccolo trucco che balza subito all'evidenza può essere quello di centrarli appositamente se ti accorgi di essere molto in negativo, risultandoti un indubbio vantaggio il raggiungere rapidamente quota 0.

Al termine dei tiri programmati, dopo il "GAME OVER" potrai conoscere il risultato che la tua bravura e i tuoi riflessi ti hanno consentito di raggiungere.

IL LISTATO

```
 1   PRINT " ◆ ƀBERSAGLI ƀ ◆"
 2   VAC
 5   INPUT "QUANTI TIRI",M
 6   IF M=0 THEN 5
10   $="12345:6789ƀ"
```

```
20   FOR N=0 TO 10
30   A$(N)=MID(N+1,1)
40   NEXT N
41   FOR N=1 TO 11 STEP 2: T(N)=10: NEXT N
50   FOR N=0 TO 10 STEP 2
60   T(N)=INT(RAN#*10)
61   PRINT CSR N; A$(T(N));: NEXT N
80   FOR R=0 TO 11
90   PRINT CSR R; " ◆";
100  IF KEY=""; PRINT CSR R; A$(T(R));: NEXT R: GOTO 80
105  IF KEY≠"" THEN 105
110  L=INT(RAN#*10): P=P+1: IF R/2≠INT(R/2) THEN 150
120  GOTO 160
150  O=O-3: IF P=M THEN 250
155  PRINT CSR R; " ƀ";: NEXT R: GOTO 80
160  IF A$(T(R))=":"; O=0: GOTO 180
170  S$=A$(T(R)): N=VAL(S$): O=O+N
180  IF P=M THEN 250
200  PRINT CSR R; A$(L);: T(R)=L: NEXT R: GOTO 80
250  FOR N=0 TO 50: NEXT N: PRINT: PRINT " ƀGAME OVER";
255  FOR N=0 TO 300: NEXT N: PRINT: PRINT "PUNTI:"; O: GOTO 2
```

Il segno ◆ nelle righe 1 e 90 è fatto con MODE . SHIFT K

(restano 47 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A$(0)..A$(10)	gli 11 tipi di bersagli
L	indica il nuovo bersaglio
M	numero di tiri stabiliti
N	valore del bersaglio colpito
O	punteggio
P	tiri effettuati
T(0)...T(11)	selezionano il carattere da visualizzare in ogni spazio

La prima parte di questo programma ti ricorderà certamente il listato di "LE MINE" ed in effetti i due giochi presentano notevoli affinità. Anche qui era necessario visualizzare nel display una serie di cifre sulle quali scorre un cursore che svolge la funzione di mirino.

La memorizzazione dei caratteri da stampare, in questo caso non solo cifre, avviene come nel precedente gioco tramite il vettore A$() i cui elementi sono preliminarmente definiti alle linee 10-40

A$(0) = "1"
A$(1) = "2"
....
....
A$(5) = ":"
A$(6) = "6"
....
....
A$(9) = "9"
A$(10)= "ƀ"

In "BERSAGLI" la stampa di tali caratteri non viene però realizzata in tutte le posizioni del pannello, ma le cifre e l'eventuale simbolo ":" sono intervallati da uno spazio e compaiono quindi solo nei posti pari.

Questo si è ottenuto con due cicli su N (linee 41/61), aventi entrambi uno STEP di 2.

Considerando che il vettore T(N) contiene i numeri (da 0 a 10) indicanti l'elemento del vettore A$() da visualizzare, nel primo ciclo si assegna il valore 10 agli elementi dispari di T(N), mentre nel secondo a quelli pari viene attribuito un valore scelto casualmente da 0 a 9.

In tal modo al momento della stampa in CSR 1,3,5,7,9,11 si visualizzerà uno spazio vuoto corrispondente appunto ad A$(10) e nelle locazioni pari potrai vedere i sei caratteri determinati dalle cifre estratte. Ad esempio T(N)=5 selezionerà l'elemento A$(5) che corrisponde ai ":".

Il cursore, rappresentato da un "♦", scorre nel display da sinistra a destra senza alcun intervento di guida da parte tua ed il suo moto avviene nel ciclo su R che ovviamente andrà da 0 ad 11.

Come nel precedente gioco delle mine anche qui, ad ogni mossa del "♦", andrà ripristinato il carattere a cui esso si era temporaneamente sovrapposto. Questo verrà quindi di nuovo visualizzato (anche nel caso si tratti dello spazio vuoto) precedendo la stampa del cursore nella successiva posizione (linee 90/100).

La velocità con cui avvengono queste due operazioni, tale da renderle praticamente simultanee, ti permette di avere una percezione quasi reale del movimento del rombo sopra i bersagli. Ma, nel caso le pile del pocket siano

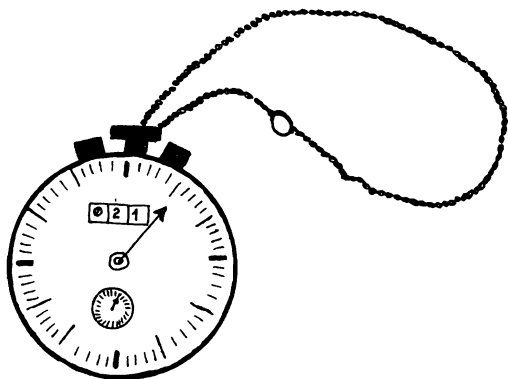
un po' consumate, ti rende anche il gioco particolarmente difficoltoso data la scarsa visibilità del mirino stesso.

Vediamo ora qualcosa che riguarda le conseguenze del lancio della tua freccia. Una volta premuto il tasto occorrerà controllare se essa ha colpito nel segno o è caduta nel vuoto. In pratica ciò corrisponde a verificare se il mirino, al momento del tiro, si trovava in uno spazio (R) pari o dispari; controllo effettuato nella linea 110 (IF R/2 ≠ INT(R/2)).

Nel caso tu non abbia abbattuto alcun bersaglio ti verranno sottratti 3 punti e si proseguirà stampando nuovamente uno spazio nella posizione in cui ti eri fermato (linee 150 e 155).

Se invece hai colpito qualcosa, sempre che non si tratti dei ":", ti verrà accreditato un punteggio pari al valore della cifra centrata.

È qui da notare (linea 170) la necessaria scissione in due istruzioni della conversione in numero di A$(T(R)); un'unica assegnazione del tipo N=VAL(A$(T(R))) avrebbe infatti provocato un ERROR 2 (vedi APPENDICE C).

IL RIFLESSOMETRO

L'idea di realizzare questo gioco ci è venuta osservando quei simpatici strumenti che misurano la velocità dei riflessi e, se non sei abbastanza svelto, inveiscono contro di te considerandoti "tardo" o "addormentato".

Vuoi mettere alla prova la tua abilità senza dover necessariamente scendere al bar sotto casa? Benissimo, con questa prova -riflessi puoi veramente dimostrare a te stesso ed agli amici di possedere una eccezionale prontezza e, cosa più importante, passare una mezz'ora di sano divertimento.

....COME SI GIOCA

Se ti sei già cimentato ed allenato a sufficienza nella prova riflessi presentata in un capitolo precedente, sei ora pronto per questo nuovo "esercizio", forse ancora meno semplice.

Qui devi veramente "mettercela tutta" perchè in ogni manche, se non raggiungi un determinato punteggio da noi adeguatamente stabilito, non puoi passare alla prova successiva e il gioco terminerà lasciandoti con un palmo di naso.

Sei pronto ad accettare ogni possibile risultato, anche la più amara delle sconfitte? Coraggio, affronta e porta a termine anche questa dura prova.

Nella prima fase, dopo la pressione di EXE, vedrai due freccette partire contemporaneamente dalle estremità del display, scorrere veloci sul pannello, incontrarsi al centro e quindi allontanarsi di nuovo verso i lati e così di seguito.

A te spetta il compito di bloccarle, premendo un tasto qualsiasi, esattamente quando esse si sovrappongono, situazione che si verifica nel sesto spazio (CSR 5).

Il punteggio che riceverai sarà tanto maggiore quanto più il movimento delle frecce è stato da te fermato in vicinanza del centro ed è così distribuito:

N° SPAZIO	PUNTI
5	200
4-6	160
3-7	120
2-8	80
1-9	40
0-10	0

Questa prima prova andrà ripetuta cinque volte e per accedere a quella seguente è necessario aver totalizzato almeno 960 punti; ciò significa che dovrai realizzare un minimo di 4 centri, cosa non semplicissima.

Un punteggio minore provoca invece la comparsa della scritta "GAME OVER", che segna la fine del gioco.

Se non è andata molto bene puoi sempre ricominciare da capo, visto che qui non hai da inserire ogni volta le duecento lire richieste invece dalle macchinette misura-riflessi dei bar.

Nella seconda prova concentra la tua attenzione sul pannello del CASIO perchè, non appena viene visualizzato il numero 999 che inizia istantaneamente a decrescere, devi premere un tasto qualunque per provocarne l'arresto. Ti verrà assegnato un punteggio uguale al numero sul quale sei riuscito a bloccare la rapidissima corsa all'indietro delle cifre.

Questa manche non presenta particolari difficoltà; l'unica nota è che il tempo di attesa prima della visualizzazione del 999 iniziale varia di volta in volta, per cui ti viene a mancare ogni riferimento.

Per poter affrontare la terza ed ultima prova devi a questo punto possedere un minimo di 1980 punti. Se fai qualche rapido calcolo puoi subito vedere che per mettere insieme un tale punteggio non ti sarà bastato realizzare nella prima fase 960 punti, ma saranno indispensabili cinque centri, corrispondenti

a 1000 punti; nella prova n.2 devi così necessariamente totalizzarne 980. Se sei stato velocissimo e accedi quindi alla parte finale del gioco puoi essere veramente fiero di te stesso.

Sul display appare il numero 000 disposto come segue:

			0		0		0				

Nella posizione più a destra, quella relativa alle unità, inizieranno subito a ruotare le cifre dallo 0 al 9 e tu, tramite la pressione di un qualsiasi tasto, devi cercare di fermarle sul 9.

La stessa operazione andrà poi ripetuta per le decine e le centinaia, che si susseguiranno ad una velocità sempre più elevata.

Il punteggio che ti viene assegnato è determinato dalla configurazione finale del numero. Ad esempio, se hai bloccato le cifre in questo modo

otterrai 979 punti. Avrai ovviamente già capito che il massimo è 999. Non ti credere però a questo punto di avere la vittoria assicurata. Infatti la tanto desiderata valutazione di "GOOD" ti verrà attribuita solo se il tuo punteggio globale ammonta a ben 2992 punti.

In questo caso hai veramente dimostrato riflessi a dir poco eccezionali che ti permetteranno senz'altro anche nella vita di far fronte ad ogni situazione che richiede prontezza e nervi d'acciaio.

IL LISTATO

```
1    VAC
2    T$="PROVA": $="GameOverƀƀƀ": F$="ƀPUNTI:":
     PRINT T$; 1: FOR C=1 TO 5
3    FOR I=0 TO 10: PRINT CSR I; "→"; CSR 10-I; "←";: L=SQR 81
4    PRINT: IF KEY=""; NEXT I: GOTO 3
5    IF KEY≠"" THEN 5
6    IF I>5; I=10-I
7    P=I*40: PRINT CSR 4; P;: GOSUB 30: S=S+P: NEXT C:
     IF S<960 THEN 35
8    GOSUB 40: PRINT 2: A=999: L=RAN#*500: GOSUB 30
9    FOR I=A TO 0 STEP -3: IF A<150 THEN 35
```

```
10    PRINT CSR 5; I;: IF KEY≠"" THEN 14
11    IF I=A-6; I=I-10: A=I-3: K=K+1
12    IF K=2; I=I-100: A=I-3: K=0
13    NEXT I
14    IF I=999 THEN 35
15    S=S+1: GOSUB 30: IF S<1980 THEN 35
16    GOSUB 40: PRINT 3, CSR 3; "0ƀ0ƀ0";: FOR I=3 TO 1 STEP -1
17    FOR C=0 TO 9: L=I: GOSUB 30: PRINT CSR 2*I; C;: IF KEY=""; 
      NEXT C: GOTO 17
18    IF KEY≠"" THEN 18
19    W=C*10↑(3-I): Y=Y+W: NEXT I: S=S+Y: L=300:
      GOSUB 30: IF S<2992 THEN 35
20    PRINT: PRINT "GOOD!ƀ"; F$; S: END
30    FOR J=0 TO L: NEXT J: RETURN
35    PRINT: PRINT $; F$; S: END
40    PRINT: PRINT F$; S;: L=300: GOSUB 30: PRINT: PRINT T$;: RETURN
```

Nella riga 3 i segni della freccia a destra e della freccia a sinistra sono fatti rispettivamente con MODE . SHIFT P e MODE . SHIFT I

(restano 0 passi)

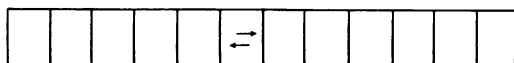
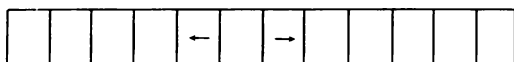
SE VUOI IMPARARE

PRINCIPALI VARIABILI	
I	determina lo spazio di stampa delle frecce e nella 2ª prova rappresenta il numero visualizzato
L	tempo casuale di attesa
P	punteggio parziale della 1ª prova
S	punteggio totale
Y	punteggio della IIIª prova

Durante la progettazione di questo programma, ancor prima di mettere nero su bianco, dubitavamo alquanto di riuscire a trasferire tutti i tre giochi della macchinetta misura-riflessi in un unico listato.

Un grosso aiuto ai nostri soliti problemi riguardanti il risparmio di passi di memoria ci è venuto dal limitato numero di istruzioni con cui abbiamo dato vita alla prima prova (linee 2/7).

La sua brevità è stata ottenuta riunendo in un unico ciclo il movimento delle due frecce aventi direzione opposta.

Durante il gioco l'illusione ottica ti ha forse tratto in inganno e può esserti sembrato che, dopo essersi incontrate al centro, le frecce cambino direzione, raggiungendo di nuovo lo spazio da cui erano partite. In realtà ognuna di esse percorre tutto il display, giungendo quindi al lato opposto a quello di partenza.

La prima freccia andrà dallo spazio 0 allo spazio 10, mentre contemporaneamente la seconda compirà il percorso inverso.

Così, nel ciclo su I da 0 a 10, esse verranno rispettivamente visualizzate nella posizione I ed in quella 10 - I.

La strana assegnazione L=SQR 81 al posto di L=9 ha lo scopo di permettere una perdita di tempo, piccola ma necessaria, fra una visualizzazione delle frecce e la successiva, senza sprecare preziosi passi di memoria. Il calcolo della radice quadrata impegna infatti il computer quasi come un breve ciclo vuoto.

La seconda prova è senz'altro quella che, richiedendo un algoritmo più complesso, ha necessitato di un maggior numero di istruzioni e di un notevole impegno.

Per creare un gioco il più possibile simile a quello osservato nei bar, si è fatto in modo che il numero da bloccare decrescesse con una rilevante rapidità. Poichè il punteggio è pari al numero che i tuoi riflessi hanno "congelato", anche una breve esitazione ti penalizza così fortemente, determinando scarsi risultati.

Pur essendo il tuo CASIO molto veloce nell'eseguire calcoli, la lentezza intrinseca del Basic non permette di decrementare il numero in questione di un'unità alla volta ad una sufficiente rapidità.

Non potevamo certo fare una prova nella quale anche una "tartaruga" ottenesse un buon punteggio e abbiamo ovviato a ciò nel modo che ora ti verrà chiarito (linee 9/12).

Il ciclo su I presenta uno STEP di -3; ciò significa che il numero visualizzato (appunto I) viene diminuito di 3 unità alla volta. Inoltre, ad ogni due decrementi, subisce una ulteriore sottrazione di 10 (IF I= A-6; I=I-10 nella linea 11) che consente un rapido cambiamento della cifra che rappresenta le decine. Così dopo il 993 compare subito il 980, con una diminuzione totale di 13.

Ma non basta ancora: al secondo calo delle decine verranno tolte in un solo colpo 100 unità (IF K=2; I=I-100 linea 12).

Ogni volta che I viene variato irregolarmente il valore di A dovrà essergli nuovamente uguagliato per poter effettuare i successivi controlli (IF I=A.....).

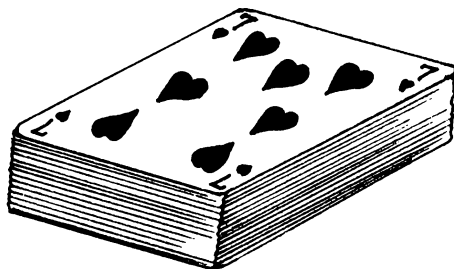
In effetti però A viene posto uguale ad I-3: questo in previsione dell'immediato STEP di -3 che avverrà al successivo NEXT.

Per una tua maggior comodità eccoti riportata la tavola delle prime visualizzazioni che ovviamente continueranno poi nel modo descritto

A	I	STAMPA	K
999	999	999	0
999	996	996	0
999	993	993	0
980	983	-	1
980	980	980	1
980	977	977	1
980	974	974	1
961	964	-	2
861	864	-	0
861	861	861	0

Per quanto riguarda l'ultima prova, l'unica cosa degna di nota ci sembra il calcolo del punteggio ad essa relativo (linea 19). Poichè esso corrisponde alla configurazione ottenuta, è evidente il diverso valore da attribuirsi alle tre cifre che la compongono (unità, decine, centinaia).

La formula W = C*10 ↑ (3 - I) esegue il calcolo necessario, moltiplicando di volta in volta la cifra fermata (C) per la giusta potenza di 10.

10^0 per le unità
10^1 per le decine
10^2 per le centinaia

Quali sono le capacità di un computer? È certo che con una sufficiente quantità di memoria i suoi limiti sono fissati soltanto dalla tua inventiva e fantasia.

In questo caso un minimo di grafica, che offre la possibilità di rappresentare i semi delle carte da gioco (♥♦♣♠), ha permesso la realizzazione del classico passatempo denominato "Guerra". Molto probabilmente esso non avrà bisogno di spiegazioni data la sua popolarità, ma se non lo conoscessi prosegui nella lettura e ti saranno dati i necessari chiarimenti.

....COME SI GIOCA

Inizialmente le 40 carte vengono distribuite tra i due giocatori, venti a testa. Alla fine dell'operazione, che richiede qualche secondo di attesa, premendo un tasto qualsiasi quattro rettangoli neri (■) lampeggeranno per tutta la durata della pressione.

Compariranno poi le due carte che il computer ha provveduto a scegliere casualmente fra quelle disponibili ad ognuno.

Tieni presente che il gioco è ideato come una gara contro il calcolatore stesso, il cui mazzo è posto alla sinistra nel pannello.

Le carte sono ordinate con valore crescente secondo lo schema 2 3 4 5 6 7 J Q K A e ad ogni mossa viene considerato vincitore il possessore della "più alta" che catturerà quella dell'avversario incrementando quindi il proprio esercito.

Esiste naturalmente la possibilità che le carte estratte risultino dello stesso valore; in questa situazione di "guerra" ne verranno automaticamente scelte due coperte e poi altre due che determineranno l'esito della manche. Il vincitore otterrà così ben tre nuovi "soldati".

Subito dopo si visualizza il numero delle carte in possesso dei due giocatori (le tue a destra e quelle del computer a sinistra). Sei quindi pronto per la prossima mano e puoi premere nuovamente un tasto che darà inizio alla successiva estrazione delle due carte.

Il gioco termina ovviamente nel momento in cui il tuo esercito o quello del computer sarà stato completamente debellato.

Quando uno dei due eserciti rimane con pochi soldati tra le sue fila al punto che una eventuale guerra regolare risulterebbe impossibile, il computer cerca di evitare questa situazione.

Nel caso non vi riuscisse la manche risulterà vinta dal giocatore più debole, permettendogli così un residuo di speranza.

Questo passatempo, come accade in realtà con vere carte da gioco, risulta talvolta piuttosto prolungato (anche più di mezz'ora); possibilità accentuata dal fatto che in questa versione una stessa carta può venire estratta più volte consecutivamente.

Non ti preoccupare se vedrai comparire il seme ♥ con maggior frequenza rispetto agli altri; ciò è dovuto alle modalità con le quali viene scelto il seme, rese necessarie dalla limitata memoria. Se possiedi il pack di espansione un buon esercizio potrà essere quello di ovviare all'inconveniente, dando a tutti i semi la stessa probabilità di uscita.

Devi inoltre prestare attenzione al fatto che non è possibile interrompere la partita, né giocare più volte di seguito senza prima aver riassegnato alle variabili $ e U$ il loro valore iniziale; questo avverrà in modo RUN.

IL LISTATO

```
$="234567JQKA"
U$="1111ƀ"

 1   FOR K=0 TO 9: A$(K)=MID(K+1,1)+U$: NEXT K
 2   N=20: O=N: Q$="1": T$="2": FOR K=1 TO N: GOSUB 50:
     NEXT K
 3   PRINT: PRINT CSR 2; N; CSR 6; O;: IF W=40; END
 4   R=2: GOSUB 70
 5   T$="0": Q$="1": GOSUB 50: GOSUB 80
 6   Q$="2": GOSUB 50: IF Y≠Z THEN 10
 7   IF W≥32; GOSUB 54: GOTO 10
 8   GOSUB 80: PRINT CSR 3; "GUERRA";: R=R+4: GOSUB 70
 9   GOSUB 50: N=N-1: Q$="1": GOSUB 50: O=O-1: GOTO 5
10   GOSUB 80: IF Y>Z; N=N+R: T$="1": GOTO 12
11   O=O+R: T$="2"
12   S=0: Q$="0": GOSUB 51: GOTO 3
50   S=INT(RAN#*10): L=VAL(Q$)
51   FOR M=S TO S+9: V=M-10*INT(M/10): $=A$(V)
52   FOR P=2 TO 5: IF MID(P,1)≠Q$ THEN 55
53   $=MID(1,P-1)+T$+MID(P+1,6-P): A$(V)=$: X(L)=V: W=
     ABS(N-O)
54   IF Q$≠"0"; RETURN
55   NEXT P: NEXT M: RETURN
70   IF KEY="" THEN 70
71   PRINT: PRINT CSR 3; "■■ƀƀ■■";: IF KEY≠"" THEN 71
73   RETURN
80   PRINT CSR L*4-1; MID(1,1);: $="♥♦♣♠"
81   PRINT CSR L*4; MID(P-1,1);: M(L)=M(L)-1: FOR K=1 TO 99:
     NEXT K: RETURN
```

Il segno ■ nella linea 71 è fatto con MODE. SHIFT Z
Il segno ♥ nella linea 80 è fatto con MODE. SHIFT J
Il segno ♦ nella linea 80 è fatto con MODE. SHIFT K
Il segno ♣ nella linea 80 è fatto con MODE. SHIFT L
Il segno ♠ nella linea 80 è fatto con MODE. SHIFT H

(restano 0 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A$(I)	elementi del vettore che contiene le 40 carte
N	numero delle carte del computer
O	numero delle carte del giocatore
W	valore assoluto della differenza tra N e O
Q$	carattere ricercato dalla subroutine 50
T$	carattere che lo sostituisce
R	numero delle carte in gioco
S	numero casuale che fornisce il valore delle carte
Y	valore della carta del computer
Z	valore della carta del giocatore

Uno dei problemi che si presentano nella stesura di un programma, forse il principale, è quello relativo alla struttura dei dati, cioè al modo in cui essi dovranno essere organizzati. In questo gioco la difficoltà consisteva nel dover prendere in considerazione ben 40 carte, ognuna distinta dalle altre per valore e seme; un uso standard delle memorie avrebbe infatti richiesto come minimo 40 variabili soltanto per la definizione di tutto il mazzo, limitando fortemente lo spazio per la stesura del programma.

L'accorgimento usato, che sfrutta la possibilità delle variabili stringa di contenere 7 caratteri, ha permesso di concentrare il tutto in un vettore di 10 elementi [A$(0)....A$(9)], ognuno dei quali formato da sei caratteri.

Eseguita la linea 1 lo schema risultante è il seguente:

A$(0) = 2 1 1 1 1 ƀ
A$(1) = 3 1 1 1 1 ƀ
A$(2) = 4 1 1 1 1 ƀ
A$(3) = 5 1 1 1 1 ƀ
A$(4) = 6 1 1 1 1 ƀ
A$(5) = 7 1 1 1 1 ƀ
A$(6) = J 1 1 1 1 ƀ
A$(7) = Q 1 1 1 1 ƀ
A$(8) = K 1 1 1 1 ƀ
A$(9) = A 1 1 1 1 ƀ

Il vettore A$ contiene tutte le informazioni necessarie.

Il primo carattere (2,3,4,...) di ogni suo elemento corrisponde ad un diverso valore della carta; ognuno dei caratteri successivi (1 1 1 1) ha una duplice funzione: la sua posizione ci indica il seme secondo la sequenza ♥ ◆ ♣ ♠ e la cifra simbolizza il possessore della carta. Ad esempio

A$(9)= A 1 1 1 1 ƀ

indica che tutti gli assi appartengono al primo giocatore.

Dovrebbe a questo punto essere chiaro che inizialmente tutto il mazzo è assegnato a questi (che risulta poi essere il computer).

In ogni variabile lo spazio presente come ultimo carattere ha soltanto lo scopo di semplificare l'uso successivo della funzione MID (linea 53).

Vediamo a questo punto in che modo avviene la distribuzione delle carte così memorizzate fra i due giocatori. Questa si effettua tramite la subroutine 50 richiamata per 20 volte con il ciclo su K nella linea 2. In essa viene scelto un numero casuale compreso fra 0 e 9 che è in relazione al valore di una carta secondo la tabella:

n° casuale	0	1	2	3	4	5	6	7	8	9
carta	2	3	4	5	6	7	J	Q	K	A

Se ad esempio è stato selezionato il numero 6, il computer prenderà in considerazione il settimo elemento del vettore [A$(6)] e porrà un "2" al posto del primo "1" che incontra (vedi linea 52)

A$(6)= " J 2 1 1 1 ƀ "

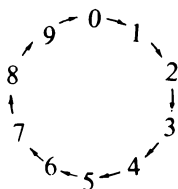
In questo modo il Jack di cuori sarà così assegnato al secondo giocatore. Questo procedimento viene ripetuto per venti volte fino alla completa suddivisione del mazzo.

Dopo le prime assegnazioni in cui tutto scorre liscio, può capitare che tutte le quattro carte di uno stesso valore appartengono già al secondo giocatore. Se

A$(5)= 7 2 2 2 2 ƀ

una ulteriore scelta del numero 5 comporterà (linea 51) la ricerca del primo "1" disponibile fra le carte restanti.

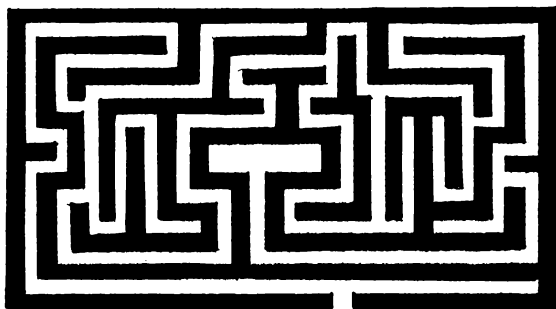
Sarà inizialmente preso in considerazione il successivo A$(6) e, nel caso, via via tutti gli altri in modo ciclico fino ad esaurire tutte le possibilità. Dopo A$(9) si riprenderà da A$(0) per terminare naturalmente ad A$(4).

Questa operazione viene effettuata tramite il ciclo su M nella linea 51 e con il calcolo presente nella 52 che esegue una così detta operazione "modulo 10", cioè fornisce il resto della divisione tra M e 10. Ciò permette di mantenere M invariato se minore o uguale a 9 e di ricollocarsi all'inizio del vettore quando M risulti maggiore.

I 544 passi disponibili sarebbero comunque risultati insufficienti a completare il programma, se non avessimo usufruito della stessa subroutine 50 ora descritta anche per l'estrazione delle carte da porre in gioco e per il trasferimento delle stesse al giocatore vincente.

Infatti con le stesse tecniche or ora descritte il computer opera la selezione di due carte, una per ogni giocatore, ponendo al posto dei corrispondenti "1" e "2" la cifra "0", indicante che la carta è attualmente in gioco.

Potrà essere utile sapere che in tutto il programma la variabile Q$ rappresenta la cifra da ricercare, mentre T$ quella che la sostituisce. Così ad esempio al termine di ogni manche verranno ricercati i caratteri 0 (Q$) presenti nel vettore per rimpiazzarli con T$ (1 o 2 a seconda del vincitore).

Il resto del programma ci sembra sufficientemente comprensibile, senza necessità di approfondimento; dovrebbe bastare la tabella delle variabili per una sua chiara lettura.

IL LABIRINTO

Questa volta ti vedo proprio male!! Sei stato rinchiuso al centro di un intricato labirinto e soltanto la tua bravura unita ad una buona dose di fortuna ed a eccellenti riflessi ti permetteranno di raggiungere incolume l'unica uscita.

Per trovarla ti è concesso soltanto un tempo assai breve e quindi eventuali errori o esitazioni ti potrebbero costare cari, condannandoti a restare prigioniero tra gli oscuri meandri.

....COME SI GIOCA

Dovrai cimentarti in un labirinto di 14x14 caselle.

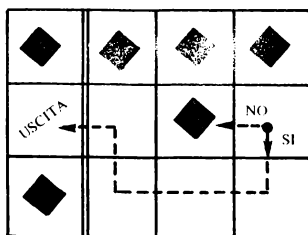
All'inizio del gioco (lanciato con RUN 8) dopo qualche secondo di attesa, necessario per la costruzione casuale del labirinto, un omino stilizzato che ti rappresenta comparirà al centro del display e tramite i tasti 6,4,8,2 dovrai cercare di uscire nel tempo massimo consentito.

TASTO	EFFETTO
6	A DESTRA
4	A SINISTRA
8	IN ALTO
2	IN BASSO

Naturalmente è possibile spostarsi soltanto negli spazi liberi.

L'omino è inizialmente al centro dello schema e ad ogni mossa viene visualizzata la riga del labirinto nella quale si trova esclusi i bordi, cosicchè non è mai possibile "vedere" dove è posta l'uscita. Al termine della prova, in caso di successo, comparirà il tempo che hai impiegato e comunque basterà premere EXE per dare il via ad un altro tentativo.

Fai attenzione, se vuoi uscire in tempo dal labirinto dovrai affrettarti a cambiare direzione ogni qualvolta la strada che hai preso risulterà bloccata dal "NO!" del computer.

Un altro consiglio: alcune volte l'uscita non è direttamente raggiungibile in linea retta ed è quindi necessario aggirare gli eventuali ostacoli (♦).

Il labirinto, sempre diverso ad ogni prova, può risultare più o meno impegnativo e il tempo necessario a trovare l'uscita varierà di conseguenza; la tua bravura ti permetterà comunque, dopo un po' di allenamento, di completare il percorso anche in situazioni piuttosto complesse. Se ti può interessare, tieni presente che il nostro record è intorno a 15.

Quando questa versione ti risulterà troppo facile, complicazioni possibili sono quelle di abbassare il tempo massimo consentito accorciando il ciclo su C nella linea 11, oppure quella di modificare la linea 12 in

```
12 GOSUB 5 : PRINT: PRINT CSR B-2; "[illegible]": G=B: H=A
```

In tal caso non si vedranno più gli ostacoli presenti nella linea visualizzata, ma l'omino continuerà a muoversi in modo da sapere in quale colonna si trova.

Una ulteriore trovata è quella di fare

```
12 GOSUB5: PRINT: PRINT CSR 6; "Ω";: G=B: H=A
```

così non si saprà neppure la colonna.

Naturalmente se modifichi la 12 è necessario allungare il ciclo su C nella 11 (inizialmente dovrebbe bastare 300).

Volendo un aiuto premi STOP e chiedi quanto valgono A e B in modo da conoscere le coordinate attuali, poi riparti premendo EXE.

IL LISTATO
(DEFM 10 RUN 8)

```
 1   PRINT: IF G=E; IF H=D; PRINT "T="; C: GOTO 8
 2   F=A: A=H: GOSUB 5: IF MID(G,1)="◆"; A=F: F$="NO":
     GOTO 4
 3   B=G: F$="BENE"
 4   PRINT CSR 5; F$;: GOTO 17
 5   $=H$(2*A-1)+H$(2*A): RETURN
 6   $=MID(1,B-1)+"ƀ"+MID(B+1,14-B)
 7   H$(2*A-1)=MID(1,7): H$(2*A)=MID(8,7): RETURN
 8   PRINT "WAIT";: FOR A=1 TO 28: H$(A)="◆◆◆◆ ◆◆◆ ":
     NEXT A: A=7: B=8: GOTO 10
 9   C=RAN#*4-2: D=SGN C: E=INTABSC: B=B+D*E: A=A+D*(E-1)
10   IF B≠14; IF B≠1; IF A≠14; IF A≠1; GOSUB 5: GOSUB 6: GOTO 9
11   D=A: E=B: A=7: B=8: FOR C=1 TO 80
12   GOSUB 5: PRINT CSR 0; MID(2,12); CSR B-2; "Ω";: G=B: H=A
13   IF KEY="6"; G=B+1: GOTO 1
14   IF KEY="4"; G=B-1: GOTO 1
15   IF KEY="8"; H=A-1: GOTO 1
16   IF KEY="2"; H=A+1: GOTO 1
17   NEXT C: PRINT: PRINT "TROPPO TARDI": GOTO 8
```

Il segno ◆ nelle linee 2 e 8 è fatto con MODE . SHIFT K
Il segno Ω nella linea 12 è fatto con MODE . SHIFT N

(restano 3 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A,B	coordinate che indicano la riga e la colonna attuali del labirinto, sia nella costruzione che in fase di gioco
C	ciclo per la durata del gioco
D,E	coordinate dell'uscita
G,H	coordinate del tentativo di mossa del giocatore
H$(I)	vettore per la memorizzazione del labirinto

Riteniamo di doverci soffermare sulla costruzione del labirinto da parte del computer. Come si può vedere nella linea 8 per memorizzarlo sono occorse le 28 stringhe H$(1)....H$(28). Ognuna contiene 7 caratteri ♦♦♦♦♦♦♦ per cui ogni riga dello schema è data dall'unione di due stringhe. Si è così inizialmente ottenuto un quadrato il cui lato contiene 14 rombi.
Nella linea 9 il computer costruisce poi il labirinto, scegliendo casualmente una direzione tra le quattro possibili (destra, sinistra, alto, basso) e quindi ponendo uno spazio al posto dell'iniziale ♦ (linea 6); così fino a che arriva ad uno dei bordi dello schema, dove verrà posta l'uscita.
La linea 9, che risulta a prima lettura piuttosto complessa, non è che l'equivalente della seguente sequenza:

```
C=INT(RAN#*4) ____________________ sceglie un numero casuale
                                   e in base a questo va in una
                                   delle 4 direzioni
IF C=0;B=B+1:GOTO 10___________ a destra
IF C=1;B=B-1:GOTO 10___________ a sinistra
IF C=2;A=A+1:GOTO 10 __________ in basso
IF C=3;A=A-1 __________________ in alto
```

Come si può osservare, anche nella linea 9 del programma quattro sono le possibilità, rappresentate dalle combinazioni di D ed E che possono assumere soltanto due valori (rispettivamente 1 e -1 e 0 ed 1). È evidente che ad esempio con E=1 la costruzione del labirinto proseguirà in senso orizzontale mentre continuerà verticalmente se E=0.

In questo modo il labirinto è costruito e risulta più o meno difficoltoso in relazione alle varie direzioni scelte casualmente, come abbiamo visto, dal computer.

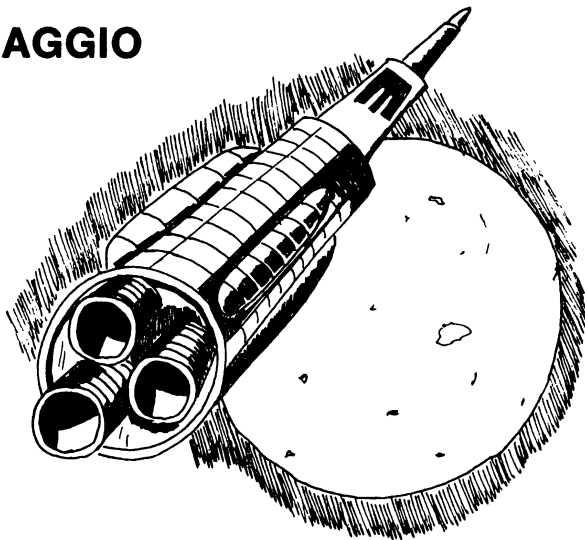
Al momento del gioco quando tu, tramite i vari tasti, effettui una mossa, il programma opera un controllo e verifica se nella posizione che dovresti occupare c'è un ostacolo (linea 2).

In questo caso devi cambiare direzione; se invece la posizione è accessibile il gioco continua e verrà eventualmente visualizzata la nuova riga del labirinto nella quale ti trovi.

Un'altra considerazione a cui il listato si presta riguarda il perchè del RUN 8 iniziale. In questo modo quasi tutte le chiamate con GOTO e GOSUB risultano seguite da un numero di una sola cifra. Questa soluzione ha permesso un risparmio di ben 9 passi, assolutamente necessari, rispetto a quella più "logica" di porre le righe 1/7 in coda al resto del programma.

Per lo stesso motivo la linea 8 va scritta in modo compatto, cioè eliminando con il tasto DEL tutti gli spazi in eccesso (quelli automaticamente inseriti tra le parole chiave e gli altri simboli) allo scopo evidente di scrivere in una sola riga ciò che altrimenti dovrebbe essere scritto in due; inoltre sono eliminate le parentesi non necessariamente richieste nella linea 9.

Ultima notazione: nella riga 2 per poter utilizzare la stessa subroutine 5 usata in precedenza, la variabile A viene salvata in F, che successivamente diventa una variabile stringa.

ALLUNAGGIO

Cape Kennedy a Parsifal!.... Cape Kennedy a Parsifal! .... Rispondete! Roger! Come vanno le cose lassù? Dispiaciuti nell'interrompere il vostro riposo, ma dobbiamo darvi una notizia poco buona. Come?... No, no, niente di preoccupante se restate calmi. Abbiamo riscontrato un malfunzionamento del modulo di comando che avete a bordo e .... Si, si, esatto! Non è più possibile controllare dalla base il vostro allunaggio: dovrete eseguirlo manualmente per non correre rischi. Si, certo! L'obiettivo è sempre lo stesso, 4 chilometri ad est del Mare della Tranquillità. Ora dobbiamo staccare, ci risentiamo tra 17 minuti esatti: in bocca al lupo! Passo e chiudo.

....COME SI GIOCA

Eccoti ora ai comandi di una navicella spaziale, pronto ad effettuare le manovre di atterraggio sulla superficie della luna, cercando di salvare la pelle dei tuoi compagni di viaggio e, ovviamente, la tua.

Quando ti siedi per iniziare la manovra sei a 30 Km dal suolo con una velocità di discesa ed una quantità di carburante che non sono fisse, ma

dipendono dal grado di difficoltà che hai scelto per compiere l'atterraggio (gli americani, si sa, sono sempre un po' matti). Infatti, dopo una breve "videata" che ti ricorda l'ordine con cui compaiono i dati, dovrai indicare, in seguito alla domanda

FORZA 1,2,3 ?

una delle cifre, che corrispondono a

1 PRINCIPIANTE
2 PROVETTO
3 ESPERTO

Non provare ad immettere una cifra maggiore di 3 perchè ti schianteresti sicuramente.

Appena effettuata la tua scelta compariranno nell'ordine velocità, quota e carburante, che varieranno rapidamente in "tempo reale". Vediamo qualche precisazione al riguardo.

La velocità è espressa in m/s ed è riferita al suolo: ciò significa che una velocità negativa indicherà la discesa della navicella, mentre un valore positivo rappresenterà la direzione verso l'alto.

-	5	3	0	■	3	0	K	■	9	7	0

Al centro del display compare ad ogni istante l'altezza dal suolo della tua capsula. Se il numero che vedi è seguito dal carattere K si tratta di chilometri; scendendo al di sotto di 1 Km tale simbolo di riferimento scomparirà ed il numero esprimerà direttamente i metri che ti separano dal suolo, facilitandoti un confronto con la velocità che, ricordiamo, è sempre in m/s.

-	1	0	5	■	5	7	2	■	1	8	0

L'ultimo numero, sulla destra del display, indica la quantità di libbre di carburante di cui puoi ancora disporre e che man mano diminuirà a seconda dei comandi dati per l'atterraggio.

Il tuo compito è infatti quello di rallentare la caduta, azionando i retrorazzi del modulo lunare e cercando di giungere al suolo con una velocità nulla o comunque molto bassa.

L'operazione sarà effettuata premendo una delle cifre (dall'1 al 9) proporzionale alla quantità di carburante che intendi consumare. Ovviamente essa sarà tanto maggiore quanto più è alta la cifra, e ciò corrisponderà ad una più sensibile diminuzione della velocità di caduta. Resta inteso che l'accensione dei retrorazzi (la pressione di una cifra) andrà effettuata ben più di una volta, variandone la potenza in relazione ai dati che compaiono nel tuo "quadro comandi".

Potrà capitarti, se freni troppo, che la velocità cambi segno, indicando una diversa direzione della navicella, ora verso l'alto: questa tecnica, utile in prossimità dell'atterraggio, può essere controproducente all'inizio. Se incorri in tale eventualità, rimarcata anche dal fatto che la quota dal suolo cresce invece di diminuire, fai attenzione a non accendere nuovamente i retrorazzi: la pressione delle varie cifre non avrà più il significato di freno, ma di ulteriore spinta verso l'alto, che renderà improbabile l'atterraggio.

L'unica cosa da fare è togliere per qualche attimo le mani dai "comandi" ed aspettare che la forza di gravità determini nuovamente una velocità negativa.

Sarà superfluo dire che una volta esaurito il carburante i tuoi "smanettamenti" non avranno più alcun esito e dovrai soltanto attendere con ansia il momento dell'impatto, sperando di essere ormai sufficientemente vicino al suolo da riuscire ad evitare il disastro.

Al termine del tentativo, se sei scampato alla distruzione, potrai avere una valutazione della tua prova, tanto più considerata quanto minore è stata la velocità di atterraggio.

VELOCITA' (m/s)	VALUTAZIONE ALLUNAGGIO
0 - 4	OTTIMO
5 - 9	BUONO
10 - 14	NORMALE
15 - 19	SCARSO
20 - 24	PESSIMO
da 25 in poi	CRAASHH!!

Comunque per tentarne un altro migliore non hai che da digitare "SI" in seguito alla scritta "RIPROVI?". Risposte diverse causeranno invece la fine del gioco.

IL LISTATO

```
FUORI PROGRAMMA IN MODO RUN:
L$ = "OTTIMO"     M$ = "BUONO"     N$ = "NORMALE"
O$ = "SCARSO"     P$ = "PESSIMO"

1    PRINT " ƀALLUNAGGIO"
2    PRINT "COMPARIRANNO VELOCITA',";"QUOTA E
     CARBURANTE";: GOSUB 30
3    V=-500: C=1000: Q=30: J=C: D=1: B$=" ƀƀƀ"
4    INPUT "FORZA 1,2,3", A: A=A*30: V=V-A: C=C-A
5    Z$=KEY: V=V-D: D=1: Q=Q+V/J: IF Q<1; Q=Q*J: J=1
6    Z$=KEY: IF Q≥1000; J=1000: Q=Q/J
7    Z$=KEY: IF Q≤0; IF J=1 THEN 13
8    PRINT CSR 1; B$; CSR 0; V; CSR 5; B$;: IF J=1000; PRINT CSR 7;
     "K";
9    PRINT CSR 4; INT Q; CSR 4; "■"; CSR 9; B$; CSR 8; C; CSR 8;
     "■";
10   IF Z$="" THEN 5
11   K=VAL(Z$): C=C-13*K: IF C<0; C=0: GOTO 5
12   D=D-10*K: GOTO 5
13   PRINT: E=INT ABS(V/5): IF E>4 THEN 16
14   PRINT "ALLUNAGGIO:";: GOSUB 30: PRINT ABS V; " ƀm/sec";
15   GOSUB 30: PRINT CSR 3; L$(E);: GOTO 17
16   PRINT "CRAASHH!!";: GOSUB 30: PRINT "CHE DISASTRO";
17   GOSUB 30: INPUT "RIPROVI", T$: IF T$="SI" THEN 3
18   PRINT "CIAO!!";: END
30   FOR I=1 TO 200: NEXT I: PRINT: RETURN
```

Il segno ■ nella riga 9 è fatto con MODE . SHIFT Z

(restano 0 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A	grado di difficoltà
C	carburante
D	variazione della velocità
J	fattore correttivo
K	potenza dei retrorazzi
L$(0)..L$(4)	valutazioni atterraggio
Q	quota
V	velocità

A questo punto, se hai letto le spiegazioni dei giochi precedenti, l'immissione di alcuni dati in modo RUN non costituisce certo una novità, e così dicasi per la loro stampa (valutazioni conclusive) in relazione alla velocità finale. Più interessante e, speriamo, istruttivo sarà invece osservare qualche particolarità sulla struttura del gioco.
Stabilito sul display lo spazio riservato alle variabili

VELOCITA' (V)	4 caratteri (3+il segno)
QUOTA (Q)	3 caratteri
CARBURANTE (C)	3 caratteri

+ 2 caratteri "■" di separazione, il gioco sarebbe risultato senz'altro troppo breve, data l'esiguità della quota di partenza (il massimo ottenibile con 3 caratteri è 999). Alcune soluzioni possibili, come diminuire lo spazio destinato al carburante o alla velocità non ci sono sembrate soddisfacenti e si è quindi adottato lo stratagemma di esprimere la quota in Km quando questa risultasse superiore alle tre cifre stabilite.

È infatti sufficientemente ragionevole pensare che il clou del gioco avvenga in prossimità dell'atterraggio, e quindi in pratica soltanto nell'ultimo tratto risulta determinante il poter seguire con precisione, metro dopo metro, l'avvicinarsi al suolo.

Risolta questa questione, almeno nella teoria, vediamo scorrendo il listato, come influiscono i tuoi comandi sul moto della navicella.

Dato per scontato l'uso del KEY al posto dell'INPUT, che renderebbe impossibile la "continuità" della discesa, avrai certo notato che la cifra

premuta viene memorizzata in Z$ e che la medesima istruzione (Z$=KEY) viene ripetuta per ben tre volte (linee 5,6,7). Questo semplicemente allo scopo di evitare che vada perduta una tua eventuale pressione del tasto durante lo svolgimento delle linee in questione.

Le istruzioni ivi contenute permettono di calcolare i nuovi valori della velocità e dell'altitudine in base al tuo precedente comando tenendo presente che, in assenza di un intervento, la navicella continuerà a scendere, attratta dalla forza di gravità.

In questo caso la variazione della velocità (V=V-D) avverrà con D=1 che rappresenta appunto l'attrazione del suolo lunare.

In base alla nota formula

$$\text{Spazio} = \text{velocità x tempo}$$

essendo la velocità espressa in metri al secondo, se supponiamo che tra ogni visualizzazione dei dati e l'altra intercorra proprio un secondo, dovrebbe risultare chiaro che ad ogni stampa lo spazio percorso è dato semplicemente dalla velocità posseduta, presa con il suo segno.

La nuova quota dovrebbe quindi essere

$$Q = Q + V$$

ma si deve tener conto che talvolta Q è espresso in metri mentre altre, come abbiamo visto, in chilometri.

Ciò porta a qualche complicazione nel listato, ed all'uso del fattore correttivo J

$$Q = Q + V/J$$

che è pari a 1000 se Q è in Km, mentre vale 1 (cioè non corregge nulla) se la quota è in metri.

Da notare inoltre che lo stesso J è utile nella determinazione del valore di Q da visualizzare, ottenendo la sua conversione in metri (moltiplicazione per 1000 nella linea 5) se inferiore ad un chilometro

```
IF Q<1; Q=Q*J
```

o la ritrasformazione in Km (divisione per 1000 nella linea 6) se, accendendo troppo i retrorazzi, hai nuovamente superato lo "spartiacque" di 999 metri

```
IF Q≥1000; J=1000: Q=Q/J
```

Diamo infine uno sguardo alle linee 8 e 9.

Esse sono relative alla visualizzazione dei risultati, e vi puoi notare come innanzitutto sia necessario cancellare i valori precedenti (mediante la stampa dei tre spazi contenuti in B$) per evitare sovrapposizioni.

Un altro accorgimento è stato quello di stampare dapprima i valori di Q e di C, e successivamente i caratteri di separazione "■" che vanno così ad occupare gli spazi altrimenti "riservati" ai rispettivi segni.

OWARE

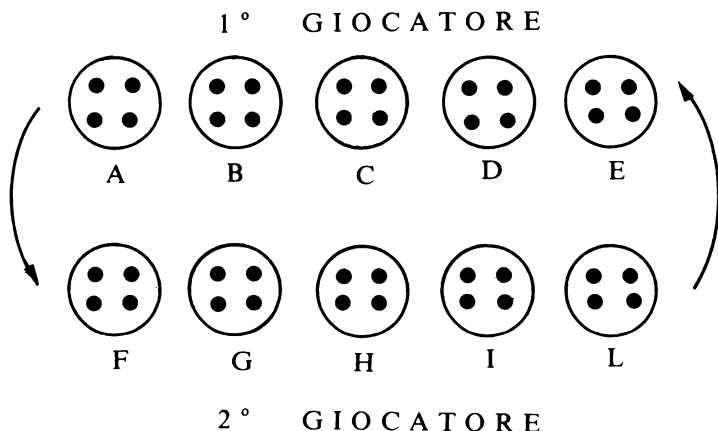
Ti sei mai chiesto come si divertono i popoli africani? Forse sarai sorpreso nel venire a conoscenza che, accanto a "passatempi" più noti, come praticare il cannibalismo, tagliare e rimpicciolire teste, danzare selvaggiamente attorno ad un falò, ve ne sono altri che sotto l'aspetto delle capacità logico-deduttive e di prontezza mentale nulla hanno da invidiare ad alcuni dei nostri giochi.

Un disinteressato consiglio: cerca di imparare a destreggiarti bene in questa gara, non si può mai sapere cosa ci riserva la vita; potresti sempre essere fatto prigioniero da una bellicosa tribù e doverti giocare la sopravvivenza contro il campione locale.

....COME SI GIOCA

Alcune tribù africane danno vita ad interessanti tornei basati su una serie di giochi molto simili fra loro, che fanno parte di antiche tradizioni. Il programma da noi presentato non è che la trasposizione per computer di uno di essi, quello che tutto sommato ci è piaciuto di più.

Si tratta di un gioco da scacchiera che si svolge fra due avversari, e qui il Casio farà soltanto da arbitro imparziale. La scacchiera è formata da 10 buche, 5 per ognuno dei contendenti ed inizialmente ogni buca contiene 4 sassolini o fagioli.

Nella realtà talvolta la scacchiera è ricavata direttamente sul terreno ed i due giocatori siedono uno di fronte all'altro, dalla parte delle buche in loro possesso.

Il gioco consiste nel prelevare tutti i sassolini presenti in una delle proprie buche a scelta, e disporli uno alla volta in ognuna delle successive, proprie e avversarie, in senso antiorario.

A questo proposito è evidente che nello schema riportato le buche indicate con E ed L sono idealmente collegate, e così dicasi per quelle A ed F.

Quando si è conclusa l'operazione di "semina" dei fagioli, a seconda di ciò che contiene l'ultima buca raggiunta, si avranno le seguenti conseguenze:

— se essa, compreso l'ultimo fagiolo ivi deposto, ne contiene 4, questi vengono prelevati fornendo così 4 punti e il gioco passa all'avversario;
— se contiene un solo fagiolo (l'ultimo depositato) la mano passa direttamente all'avversario;
— con un numero di fagioli diverso da 1 e da 4 il medesimo giocatore dovrà prelevarli, continuando a disporli nelle buche successive fino a che non si ricada in una delle prime due alternative (buca con 1 o 4 fagioli).

Non eccessivamente complicato, vero? Ma giocarlo correttamente con intelligenza, senza fare mosse a "casaccio", è un altro discorso. Prima di fare un esempio concreto vediamo come è avvenuta la conversione per il tuo pocket.

I due contendenti dovranno inserire il proprio nome (ricorda, non più lungo di 7 caratteri) e subito dopo comparirà il piano di gioco.

Ovviamente le 10 buche (o meglio le cifre che ne quantificano il loro contenuto) sono visualizzate una di seguito all'altra, convenendo che le prime cinque appartengano al primo giocatore, le altre al suo avversario. Si dovrà quindi fare particolare attenzione ad indicare la casella dalla quale si intendono prelevare i sassolini (o i fagioli), anche perchè esse sono numerate da 0 a 9, in modo che la scelta possa essere stampata nello spazio alla estrema destra del display (al posto del ?).

Ricapitolando, quando deve muovere il primo giocatore, egli potrà indicare soltanto le buche dalla 0 alla 4, mentre il secondo disporrà di quelle 5...9. Il Casio ti facilita molto, indicando ogni volta colui che è di turno, e ti impedisce di fare errori di input, che non verranno accettati; sarebbe comunque meglio che ti costruissi una mascherina, magari con due colori, da inserire temporaneamente sopra la tastiera in modo da evitare confusione differenziando opportunamente le due serie di "buche".

mascherina →

Una volta indicata quella prescelta (senza premere EXE) risulta tutto automatico, e non dovrai far altro che osservare sul display come avviene la tua "semina" che risulterà da sinistra a destra, con la casella n° 9 idealmente collegata alla n° 0; se hai ottenuto un buon "raccolto" i 4 punti guadagnati ti saranno assegnati e comunque verrà visualizzata la situazione attuale dei contendenti. Potrebbe verificarsi, e senz'altro prima o poi lo vedrai, che in qualche buca vi sia un numero di sassolini maggiore di 9; a questo proposito abbiamo usufruito anche delle lettere alfabetiche, considerate come naturale proseguimento delle 10 cifre.

Così se vedrai comparire il carattere C significa che in quella casella sono presenti 12 sassolini; eccoti comunque per tua comodità la tabella completa

lettera alfabetica

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

numero che simbolizza

La gara viene fatta terminare quando un giocatore rimane con tutte le proprie buche prive di sassi e per questó talvolta la conclusione, anche se scontata, si prolunga per qualche mossa in più del dovuto.

Quando sei in vantaggio una buona tecnica consiste nel cercare di sfruttare questo fatto, tentando di far cadere tutti i sassi nelle caselle avversarie in modo da giungere rapidamente alla conclusione. Quest'ultima è indicata dalla comparsa della parola FINE, a cui seguirà, premendo EXE, il risultato acquisito.

Essendo questo gioco ancora poco conosciuto terminiamo con un esempio concreto di un suo parziale sviluppo, con l'analisi di alcune alternative di mossa. Data questa situazione

2	0	3	1	3	0	2	5	A	2	■	?
0	1	2	3	4	5	6	7	8	9		
SINISTRA					DESTRA						

— Il giocatore di SINISTRA muove la buca 0 e semina 2 sassi alle buche 1 e 2. Al termine la buca 2 contiene 4 sassi, che vengono prelevati: SINISTRA ottiene così 4 punti ed il gioco passa all'avversario.

— Il giocatore di SINISTRA muove la buca 2 e semina 3 fagioli alle buche 3, 4, 5. Al termine la buca 5 contiene 1 fagiolo, ed il gioco passa direttamente all'avversario.

— Il giocatore di DESTRA muove la buca 6 e semina due sassi alle buche 7 e 8. Al termine la buca 8 contiene B(=11) sassi, quindi DESTRA li raccoglie e li semina nelle buche 9,0,1,2,3,4,5,6,7,8 e 9. A questo punto la buca 9 contiene 4 sassi che, prelevati definitivamente, forniscono 4 punti a DESTRA.

IL LISTATO

FUORI PROGRAMMA IN MODO RUN:
$ = "0123456789ABCDEFGHILMNOPQRSTUV"

```
1    FOR K=1 TO 2: INPUT "NOME",L$(K): V(K)=0: NEXT K
2    FOR K=0 TO 9: A$(K)="4": NEXT K: T=1
3    GOSUB 50: GOSUB 51
4    GOSUB 60
5    Z$=KEY: IF Z$="" THEN 5
6    Y=VAL(Z$): IF T=1; IF Y>4 THEN 5
7    IF T=2; IF Y<5 THEN 5
8    PRINT CSR 11; Z$;
```

```
9     GOSUB 51: GOSUB 70
10    P=R-1: IF P=0 THEN 4
11    GOSUB 30: L=0
12    Y=Y+1: IF Y=10; Y=0
13    GOSUB 70: A$(Y)=MID(R+1,1): PRINT CSR Y; A$(Y);: GOSUB 51
14    L=L+1: IF L<P THEN 12
15    IF A$(Y)="4"; V(T)=V(T)+4: GOSUB 30: GOTO 17
16    IF A$(Y)≠"1"; GOSUB 51: GOTO 9
17    FOR U=0 TO 5 STEP 5: FOR S=U TO U+4
18    IF A$(S)≠"0" THEN 20
19    NEXT S: GOSUB 51: PRINT: PRINT "FINE": GOSUB 21: END
20    NEXT U: GOSUB 21: T=3-T: GOTO 3
21    FOR V=1 TO 2
22    PRINT: PRINT L$(V); ":"; V(V);: GOSUB 51
23    NEXT V: RETURN
30    A$(Y)="0": PRINT CSR Y; "0";: RETURN
50    PRINT: PRINT "GIOCA ␢"; L$(T);
51    FOR Q=1 TO 100: NEXT Q: RETURN
60    FOR K=0 TO 9: PRINT CSR K; A$(K);: NEXT K
61    PRINT CSR 10; "■?";: RETURN
70    FOR R=1 TO 30: IF A$(Y)≠MID(R,1); NEXT R
71    RETURN
```

Il segno ■ nella riga 61 è fatto con MODE . SHIFT Z

(restano 2 passi)

SE VUOI IMPARARE

PRINCIPALI	VARIABILI
A$(0)..A$(9)	fagioli presenti nelle 10 buche
L	effettua "manualmente" un ciclo fino a P
L$(1) , L$(2)	nomi dei giocatori
P	numero dei fagioli presenti nella buca Y
T	indica il giocatore di turno
V(1) , V(2)	punteggi dei giocatori
Y	buca selezionata dai contendenti

Se la disponibilità di memoria lo consente, è sempre ottima cosa fare in modo che l'utilizzatore, durante lo svolgimento di un programma, non possa premere tasti indesiderati che causano risultati non previsti o addirittura

errori che costringono a ripetere l'esecuzione. Spesso non abbiamo potuto realizzare tutto quanto si dovrebbe, ma un minimo di controllo è veramente indispensabile.

In questo gioco, ad esempio, la leggera difficoltà di "lettura" determinata dal dover numerare le prime cinque buche dallo 0 al 4 (anzichè dall'1 al 5) potrebbe indurre qualcuno un po' distratto ad una svista.

Nelle linee 6 e 7 si è quindi fatto in modo che il primo giocatore (T=1) non possa premere una cifra maggiore di 4, mentre per il secondo (T=2) si esegue il test sulla condizione "minore di 5".

Naturalmente se invece di un numero premi un altro tasto, il programma va in "tilt", ma non si è potuto fare diversamente.

Già che abbiamo nominato la variabile T ti facciamo subito notare un piccolo accorgimento per indicare il cambio di turno fra i due giocatori. Quando uno di questi ha terminato la propria mossa, verificate le condizioni di proseguimento della gara (linee 17/20), nella linea 20 troviamo la semplicissima istruzione T=3-T.

In tal modo se T era 1 diventa uguale a 2 e viceversa, determinando così lo scambio voluto.

Ma passiamo ora a qualcosa di più "difficile" (naturalmente stiamo scherzando), relativa all'algoritmo vero e proprio.

Avrai già visto che tutti i "fagioli" presenti in ogni buca sono memorizzati nel vettore A$(0)...A$(9), i cui elementi all'inizio della gara hanno valore 4. Terminata la fase di immissione del numero relativo alla buca richiesta (Y) si passa alla subroutine 70, dove troviamo una "variazione sul tema" dell'estrazione di caratteri da $. Tale variabile viene percorsa fino a che non vi si rintraccia il posto occupato dal numero di "fagioli" presenti nella buca Y.

Se ad esempio nella buca n°3 vi sono A(=10) fagioli, poichè tale lettera è all'undicesimo posto in $, si uscirà dal ciclo (ritornando al programma principale) con R=11. È evidente allora che il numero di fagioli ricercato (10) è uguale ad R-1 e tale valore è memorizzato nella variabile P.

Dopo aver controllato un tuo anche involontario tentativo di barare scegliendo una buca vuota (IF P=0 THEN 4), nella subroutine 30 vengono prelevati i fagioli presenti (ponendovi il carattere "0") e poi si procede alla semina (linee 12/14).

Puoi innanzitutto notare come il ciclo fino a P sia qui effettuato per così dire "manualmente" incrementando la variabile L, anzichè adoperare un FOR...NEXT.

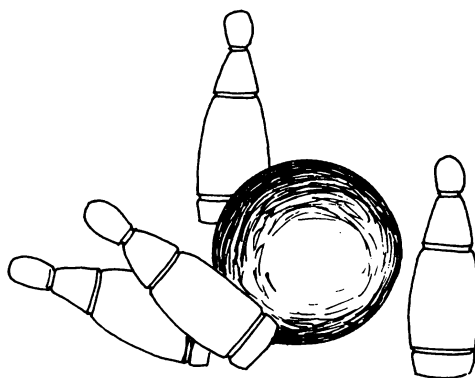
Questo perchè le precedenti e ripetute uscite dal ciclo su R prima della sua logica conclusione creavano qualche problema (vedi appendice C).

Nelle linee in questione, per un numero di volte pari a P (IF L<P THEN 12), si mette un fagiolo in ognuna delle buche successive incrementando il loro valore. Per la presenza anche delle lettere ciò viene fatto operando direttamente sui caratteri anzichè con i numeri.

Ricordando che dopo la n° 9 si ricomincia dalla n° 0 (IF Y=10;Y=0) per ogni buca viene dapprima stabilito qual è il carattere attualmente presente (sempre nella subroutine 70) e quindi esso viene semplicemente sostituito con quello successivo (A$(Y)=MID(R+1,1)).

Mentre le conseguenze del tiro ci sembrano facilmente comprensibili diciamo ancora due parole sulla verifica della eventuale conclusione del gioco (linee 17/20).

I due cicli annidati controllano se, per cinque buche consecutive (FOR S=U TO U +4) a cominciare da quelle n° 0 e n° 5 (FOR U=0 TO 5 STEP 5), in ciascuna di esse è presente il carattere "0".

Se ciò è falso si prosegue con il cambio di turno già descritto, altrimenti la gara risulterà conclusa.

Una breve e ripida scaletta, luci azzurre soffuse, una moltitudine di ragazzi con le mani incollate ai diversi videogame appoggiati alle pareti: l'inconfondibile ritratto di una sala-giochi. Sulla sinistra un fornitissimo bar, mentre sulla destra, in un ampio salone riservato, otto smaglianti piste in parquet di legno, con 10 birilli in fondo ad ognuna di esse. È il bowling.

Sei un po' emozionato prima di salire sulla pedana? Cerca di controllarti e attendi il tuo turno con calma. E, mi raccomando, non fare proprio una "magra"; sforzati di avere una mira almeno decente.

....COME SI GIOCA

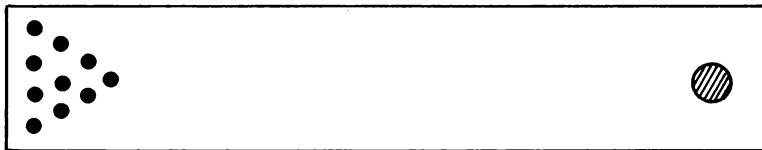
Se hai già avuto modo di giocare a bowling questo è proprio il programma adatto, che ti permette di allenarti a dovere anche se non hai la possibilità di recarti al palasport o alla sala-giochi. Non hai mai praticato questo sport, anzi non sai neppure di cosa si tratta? Ecco allora l'occasione per cominciare e, se non le conosci, per dare uno sguardo alle regole, da noi riassunte brevemente.

Si tratta di abbattere il maggior numero possibile di birilli lanciando una palla (nella realtà piuttosto pesante) per i dieci turni che ogni giocatore ha a disposizione.

In ognuno di essi verranno effettuati uno o due tiri e il punteggio sarà in generale pari al numero dei birilli abbattuti con alcune regole particolari. Se con la prima palla si abbattono tutti i birilli si ottiene uno STRIKE, non si effettua il secondo lancio e ai 10 punti già incamerati verranno aggiunti quelli dei due tiri seguenti. Se invece soltanto una parte dei birilli viene abbattuta si effettua il secondo tiro e nel caso cadano tutti quelli rimasti si realizza uno SPARE, che consentirà di aggiungere ai tuoi 10 punti anche quelli del tiro successivo.

Nel programma presentato tutte queste caratteristiche rimangono, e il tuo Casio calcolerà automaticamente il punteggio totale comprensivo degli eventuali SPARE o STRIKE.

L'unica limitazione è quella di non consentirti tiri supplementari nel caso siano ottenuti nell'ultimo turno di gioco; ma vediamo praticamente di cosa si tratta.

I 10 birilli sono disposti su quattro file, secondo lo schema

e così, per dieci volte, sul display comparirà sulla sinistra il numero dei birilli da abbattere per ogni fila e al centro l'indicazione di quale turno stai effettuando.

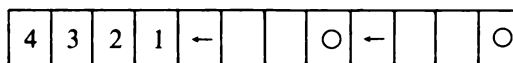
Dopo qualche attimo di attesa la suddetta indicazione scomparirà e sulla destra cominceranno a ruotare rapidamente le cifre dall'1 al 9, rappresentanti simbolicamente la palla che si muove veloce in senso trasversale da un lato all'altro della pista.

Il tuo compito sarà quello, premendo un tasto qualsiasi, di far partire la palla quando questa si trova al centro, ed avere quindi maggiori probabilità di colpire tutti i birilli.

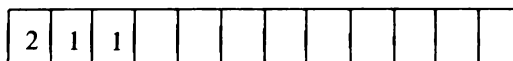
In pratica dovrai cercare di fermare la rotazione delle cifre sul 5, esattamente a metà fra l'1 ed il 9, che rappresentano quindi i bordi della corsia di lancio.

Ti accorgerai presto che non sempre, anche bloccando sul 5, si ottiene STRIKE o SPARE: ciò significa che, pur lanciando la palla esattamente al centro, essa ha acquistato lungo il percorso strani effetti che l'hanno costretta a deviare dalla traiettoria ideale, come spesso succede ânche giocando nella realtà.

Non appena avrai premuto un tasto la tua palla comparirà sulla destra del pannello, dirigendosi verso i birilli

4	3	2	1	←			O	←			O

e abbattendone un certo numero

birilli
ancora
in piedi

2	1	1									

A seconda di quanto realizzato, per le regole del gioco, dovrai ora effettuare il 2° tiro o passerai di diritto al turno successivo se comparirà la scritta "STRIKE".

Al termine della partita, premendo EXE, potrai conoscere il punteggio ottenuto.

Forse è inutile farlo notare ma è ovvio che, oltre che da solo, puoi giocare in competizione con quanti avversari desideri. In tal caso non si potrà effettuare un turno a testa, ma sarà necessario che ognuno completi la propria partita prima di passare il Casio all'altro giocatore.

IL LISTATO

```
1    VAC
2    PRINT "ƀƀBOWLING": $="4321": FOR K=1 TO 10: X=10
3    FOR I=0 TO 3: A$(I)=MID(I+1,1): PRINT CSR I; A$(I);: NEXT I
4    PRINT CSR 6; K; "]"; CSR 6; "[";: GOSUB 50: PRINT CSR 6;
     "ƀƀƀƀ";
5    E=E+1: R=X
6    FOR J=1 TO 9 STEP .8: PRINT CSR 10; INT J;: IF KEY=""; NEXT J:
     GOTO 6
```

```
7    Q=4-ABS(J-5): FOR I=0 TO 3: V$=A$(I): IF V$="" THEN 12
8    W=VAL(V$): Z=INT(RAN#*W+2.5)-Q: IF Z>W; Z=W
9    IF Z≤0; A$(I)="": GOTO 12
10   FOR P=1 TO 4: H$=MID(P,1): IF VAL(H$)≠INT Z; NEXT P
11   A$(I)=H$
12   NEXT I: FOR P=11 TO 0 STEP -1: PRINT CSR P; "○";: GOSUB 51
13   PRINT CSR P; " ƀ";: IF P≤3; PRINT CSR P; A$(P);
14   NEXT P: FOR P=0 TO 3: H$=A$(P): IF H$≠""; X=X-VAL(H$)
15   NEXT P: T=T+X: IF Y≠0; T=T+X: Y=Y-1
16   X=R-X: IF E=1; IF X≠0 THEN 5
17   IF E=1; PRINT CSR 3; "STRIKE";: Y=Y+2: GOTO 19
18   IF X=0; PRINT CSR 3; "SPARE";: Y=Y+1
19   E=0: GOSUB 50: PRINT: NEXT K: PRINT "TOTALE="; T: END
50   FOR S=1 TO 200: NEXT S
51   RETURN
```

Nella riga 4 e i segni] e [sono fatti con MODE . SHIFT Y, T
Nella riga 12 il segno ○ è fatto con MODE . SHIFT A

(restano 6 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A$(0)...A$(3)	birilli da abbattere per ogni linea
E	numero del lancio (1° o 2°) ad ogni turno
Q	abilità nel tiro
R	totale dei birilli da abbattere
T	punteggio
W	n° dei birilli in ogni linea prima del lancio
X	birilli da abbattere/birilli abbattuti
Y	n°degli abbuoni da accreditare
Z	birilli rimasti per ogni linea

Un gioco forse tra i più difficili, questo Bowling, ed anche noi stessi, a qualche mese di distanza dalla sua "creazione", abbiamo dovuto faticare un poco per comprendere nei dettagli il listato. Vediamolo insieme.

Le prime linee non dovrebbero costituire un problema: vi sono dieci turni di gioco (e quindi un ciclo su K da 1 a 10) in ognuno dei quali devi abbattere

dieci birilli (X=10) e il numero di questi in ciascuna linea (4,3,2,1) viene memorizzato nel vettore A$(I). Sulle assegnazioni della linea 5 torneremo in seguito, mentre è evidente che lo STEP .8 della riga seguente ha lo scopo di fornire la "giusta" velocità alla rotazione delle cifre sulla destra del display. Già meno chiare sono le successive istruzioni, in particolare le assegnazioni di Q e di Z: cerchiamo di scoprire da quale ragionamento esse derivino.

La quantità dei birilli abbattuti deve dipendere in modo determinante dall'attimo in cui fai partire la palla, quindi dalla cifra che resta bloccata alla pressione di un tasto.

Se indichiamo con Z il numero dei birilli rimasti (per ogni linea), tale variabile avrà quindi due componenti: una (Q) data dalla tua abilità, ed una dovuta al caso. Quest'ultima si è fatta dipendere ovviamente dal numero dei birilli presenti su una determinata linea prima del lancio (W) con una successiva correzione data dal valore 2.5 trovato sperimentalmente in modo che la fortuna incida nella giusta misura.

La componente di abilità Q dovrà essere simmetrica rispetto al centro della pista (cifra bloccata sul 5) essendo evidente che ai fini della buona riuscita del lancio è indifferente fermarsi sul 4 o sul 6. Il calcolo Q=4-ABS(J-5) serve appunto ad assegnare a Q un valore compreso fra 0 e 4 a seconda della maggiore o minore lontananza dal centro

cifra che si visualizza	suo effettivo valore (J)	componente di abilità (Q)
1	1	0
1	1.8	0.8
2	2.6	1.6
3	3.4	2.4
4	4.2	3.2
5	5	4
5	5.8	3.2
6	6.6	2.4
7	7.4	1.6
8	8.2	0.8
9	9	0

È altresì perfettamente logico fare in modo che, per qualsiasi risultato, Z non possa mai superare il numero W dei birilli presenti su una determinata riga

IF Z>W; Z=W   nella linea 8

La comprensione della routine relativa alla palla che si dirige verso il bersaglio sarà a questo punto del libro sicuramente una formalità (linee 12-13), così come il rimando ad una subroutine vuota (GOSUB 51) che serve solo a perder tempo tanto quanto basta. Leggermente più "delicate" sono le linee 9,10 e 11, dove avviene la memorizzazione nel vettore A$(I) del numero dei birilli rimasti in ogni linea (Z) dopo il tuo lancio.

Se $Z \leq 0$ si è preferito visualizzare uno spazio vuoto anzichè lo "0" mentre per gli altri valori si è utilizzato il ciclo su P che trasforma il valore di Z in una variabile di carattere e consente di assegnare ad A$(I) il segno "1" anche nell'ipotesi $0<Z<1$.

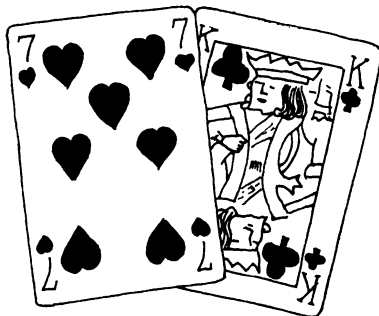
Infatti nel caso di completamento infruttuoso di tutto il ciclo verrà attribuito ad A$(I) l'ultimo carattere assunto da H$, cioè MID(4,1) che è appunto "1".

Sarà opportuno spendere inoltre qualche parola sulla tecnica da noi escogitata per compattare il calcolo del punteggio, complicato dalla necessità di tener conto degli eventuali abbuoni in caso di STRIKE o SPARE. Ovviamente con maggiore disponibilità di memoria non ci sarebbero stati problemi, ma così bisogna "arrangiarsi".

Abbiamo quindi pensato di incrementare una variabile (Y) ogni volta che verranno abbattuti tutti i birilli; tale incremento sarà di 2 se ciò avviene al primo lancio (E=1), mentre facendo l'"enplein" in due tiri (E=2) si avrà soltanto Y=Y+1.

Dovresti a questo punto essere in grado di comprendere anche le ultime linee del listato, osservando come al punteggio accumulato fino a quel momento (T) venga aggiunto di volta in volta il numero dei birilli abbattuti (X) calcolato nel ciclo su P della linea 14. Se la variabile Y è positiva la somma T=T+X sarà ripetuta, fornendoti in questo modo i punti di abbuono che ti spettano.

Un particolare a cui devi prestare attenzione è che al termine di quanto detto la variabile X conterrà nuovamente il numero dei birilli rimasti (anzichè quelli abbattuti), mediante la sottrazione da R in cui nella linea 5 era stato salvato il valore originale di X.

SETTE E MEZZO

Il tuo compagno di giochi è a letto con una fastidiosa influenza o è particolarmente impegnato nello studio per l'interrogazione di domani? Se ti senti troppo solo è la giornata giusta per raccogliere la sfida che ti viene lanciata dal computer e cercare di sconfiggerlo in questo conosciutissimo gioco di carte.

Potresti anche spuntarla, avvantaggiato come sei dal fatto di gareggiare per secondo; ma stai attento perchè anche così hai di fronte un avversario di tutto rispetto, che potrebbe meritarsi un bel "7 e $^{1}/_{2}$".

....COME SI GIOCA

Come già si può intuire dal nome stesso, in questo simpatico passatempo, in realtà un vero gioco d'azzardo, i partecipanti devono riuscire ad ottenere un punteggio che si avvicini il più possibile a 7 e mezzo.

Per formare questo numero si continua nella richiesta di carte, finchè si ritiene che la somma dei loro valori sia abbastanza vicina alla cifra da realizzare, così da evitare un'altra chiamata.

Infatti il rischio maggiore è quello di "sballare", cioè di avere un punteggio che supera il tanto desiderato 7 e $^{1}/_{2}$; quando ciò accade la manche è senz'altro persa a meno che, ovviamente, anche gli altri avversari non abbiano sballato.
Se, cosa molto improbabile, sei l'unica persona al mondo completamente all'oscuro di questo famoso gioco ti sarai forse domandato come sia possibile, sommando i valori delle carte richieste, realizzare 7 e mezzo con quel mezzo punto di troppo; e anche se conosci già tutto non ti annoiare alle nostre spiegazioni: una rinfrescatina alle idee non fa mai male.

Ebbene, quel 0,5 deriva dal fatto che alle figure (Jack, Queen e King) è stato assegnato un valore pari proprio a mezzo punto.
Così queste carte:

ti daranno un punteggio di 4,5 mentre

4♠ 2♦ J♥ 4♣

ti faranno "sballare" poichè danno un totale di 10,5.
C'è però un'altra eccezione che viene a complicare un po' le cose e riguarda la donna di cuori, la cosiddetta "matta".
Cosa succede quando si "pesca" questa fortunatissima carta? Ad essa, al posto dello scontato 0,5 può essere attribuito qualsiasi altro valore, purchè intero. Potrà quindi di volta in volta rappresentare il numero desiderato (dall'1 al 7), intervenendo praticamente come un jolly. Un esempio?
Se hai in mano una figura ($^{1}/_{2}$ punto) ed alla succesiva richiesta ti ritrovi con la "matta", attribuirai ovviamente a quest'ultima il valore di 7, per formare il vincente 7 e $^{1}/_{2}$.

K♣ + Q♥ = 7,5
/
la matta

Così la regina di cuori ti consente sempre di realizzare almeno sette punti.
Tutte le regole che abbiamo fin qui chiarito si ritrovano pari pari nella nostra versione, nella quale hai come avversario nientemeno che il tuo amico computer.
È sempre lui il primo a "pescare" casualmente una carta tra le quaranta del mazzo; carta che a te rimane però nascosta dietro un "■" posto all'estremità

sinistra del display. Dopo che il CASIO ha considerato la possibilità di proseguire o meno, ti renderà nota la sua decisione visualizzando sul pannello la scritta "ANCORA" o quella opposta di "BASTA".

Nel primo caso provvederà subito ad estrarre dal mazzo una nuova carta che anche tu potrai vedere stampata nel display e così ancora finchè ritiene sia giunto il momento di cederti la mano.

È piuttosto evidente che tu sei avvantaggiato non di poco in quanto hai modo di prendere in visione le "uscite" dell'avversario.

Calcolando la somma dei loro valori avrai un'idea, anche se non esatta mancando il primo, di quello che egli può aver realizzato e giocare quindi di conseguenza. Talvolta, se hai un po' di fortuna, potrai addirittura vedere un suo eventuale sballo.

Dopo che la tua carta è comparsa automaticamente, devi rispondere alla domanda "ANCORA?" digitando secondo la necessità il"SI" o il "NO". Il gioco continua fino a quando prosegui nelle successive richieste; appena queste saranno cessate si visualizzano i punteggi ottenuti e saprai quindi chi, tra voi due, si è avvicinato di più al difficile traguardo del 7 e $^1/_2$ ed è così il vincitore.

Per cimentarti in un'altra gara ti basta premere il tasto EXE e metterai nuovamente alla prova la tua fortuna e le tue capacità deduttive.

Ti può interessare sapere che nelle nostre partite la percentuale delle vittorie del CASIO si aggira intorno ad 1/3 delle manche effettuate; ciò significa che anche tu, con un po' di scaltrezza e concentrazione, potrai riuscire a sconfiggerlo, soprattutto in una gara a lungo termine, costituita almeno da una ventina di tentativi.

IL LISTATO

```
 1   PRINT "SETTEeMEZZO"
 2   VAC
 3   $="A234567JQK": FOR A=1 TO 10: A$(A)=MID(A,1)+
     "♥ ♦ ♣ ♠ ƀ": NEXT A
 4   GOSUB 80: PRINT: PRINT "■";: GOSUB 90: S=S+1
 5   PRINT CSR 4; M$;: GOSUB 70: IF M$≠"BASTA"; GOSUB 80:
     GOSUB 90: GOTO 5
 6   PRINT "ORA STA A TE";: GOSUB 70
 7   GOSUB 80: IF L=9; IF P=2; V=1
 8   Y=Y+L: IF L>7; Y=Y-L+.5
 9   IF Y≥8; PRINT "HAI SBALLATO";: GOTO 13
10   INPUT " ƀƀANCORA", $: IF $="SI" THEN 7
11   IF V=1; Y=7.5-FRAC Y
```

```
12    PRINT "TU HAI"; Y;
13    GOSUB 70: PRINT "IO HO";: IF X≥8; PRINT " ↑SBALLATO":
      GOTO 2
14    PRINT X: GOTO 2
70    FOR R=1 TO 400: NEXT R: PRINT: RETURN
80    L=INT(RAN#*10)+1: P=INT(RAN#*4)+2
81    $=A$(L): IF MID(P,1)="1" THEN 80
82    IF S≠0; PRINT: PRINT MID(1,1); MID(P,1);
83    $=MID(1,P-1)+"1"+MID(P+1): A$(L)=$: RETURN
90    X=X+L: IF L>7; X=X+.5-L
91    IF L=9; IF P=2; X=7.5-FRAC X
92    M$="BASTA": IF X≤6; IF X↑3≤RAN#*441; M$="ANCORA"
93    GOSUB 70: RETURN
```

I segni ♥ ♦ ♣ ♠ nella riga 3 sono fatti con MODE . SHIFT J, K, L, H
Il segno ■ nella riga 4 è fatto con MODE . SHIFT Z

(restano 5 passi)

SE VUOI IMPARARE

PRINCIPALI VARIABILI

A$(1)...A$(10)	vettore utilizzato per la memorizzazione delle 40 carte
L,P	numeri casuali che selezionano il valore e il seme della carta da immettere in gioco
M$	contiene a seconda dei casi la stringa "BASTA" o quella "ANCORA"
Y,X	rispettivamente il tuo punteggio e quello del CASIO

La tecnica adottata per memorizzare le quaranta carte da gioco è analoga a quella vista nel precedente "Guerra".

Anche qui si è utilizzato un vettore di 10 elementi (A$(1)...A$(10)) ognuno

dei quali contiene le quattro carte di un determinato valore, con questa convenzione:

A$(1) = A ♥♦♣ ♠ ƀ
A$(2) = 2 ♥♦♣ ♠ ƀ
A$(3) = 3 ♥♦♣ ♠ ƀ
....
....
A$(10)= K ♥♦♣ ♠ ƀ

Diversamente da come avveniva in "Guerra" abbiamo però potuto inserire direttamente all'interno del vettore i quattro semi, facilitando le successive operazioni di stampa.

La scelta delle carte che entrano in gioco, sia per te che per il computer, è effettuata nella subroutine 80 che ne fornisce dapprima il valore (L) e quindi il seme (P).

Potrai qui notare una maggiore linearità dell'algoritmo rispetto al precedente gioco di carte, dovuta a diversi fattori.

Innanzitutto non occorre qui suddividere il mazzo tra i due partecipanti, nè distinguere da chi la carta è stata prelevata. In ogni caso verrà posto il carattere "1" nella locazione selezionata ad indicare che, per quella manche, non sarà più possibile ripetere la medesima estrazione (linea 83).

A$(4) = 4 ♥ 1 ♣ 1 ƀ

stabilisce ad esempio che i quattro di quadri e di picche sono attualmente in gioco.

Un altro elemento di semplificazione del listato è senz'altro la non interferenza tra le tue mosse e quelle del CASIO ed ancora la possibilità di scegliere tutte le carte necessarie tramite due sole funzioni RAN#. Ciò non era così semplice in "Guerra" dove, in una situazione di grande disparità di forze, la ricerca completamente casuale di una carta appartenente al giocatore più debole avrebbe richiesto un tempo assai prolungato.

Qualche complicazione in più ci viene invece dal calcolo dei punteggi (X per il computer ed Y per te).

Come ben sai ogni carta estratta dà un punteggio pari al suo valore, tranne le figure che valgono tutte mezzo punto. Così in tale eventualità, cioè se L è maggiore di 7, ad X o ad Y viene aggiunto 0,5 e sottratto il valore di L, precedentemente attribuito (linee 8 e 90). Ad esempio, se hai solo un Jack (L=8), il tuo punteggio sarà 8 + 0,5 -8 cioè proprio mezzo punto.

Mentre per quanto ti riguarda ciò è sufficiente in quanto sarai tu stesso a decidere se proseguire o meno nella ricerca del 7 e mezzo, abbiamo dovuto invece dotare il CASIO di una pur semplicissima strategia di gioco.

Operando sempre per primo esso non potrebbe comunque trarre vantaggio dalla visione delle tue carte e quindi si è pensato di fargli senz'altro cessare la richiesta se possiede già più di 6 punti. Diversamente il suo eventuale proseguimento sarà stabilito dal caso, ma tanto più probabile quanto minore è il suo punteggio (linea 92).

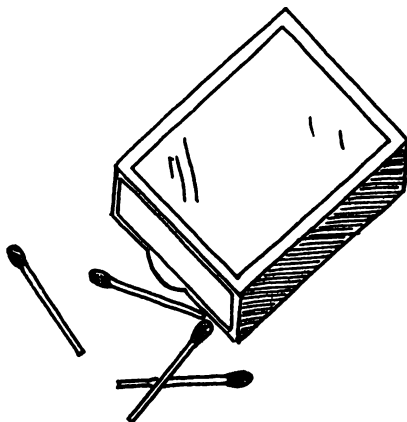
La formula riportata dà al CASIO all'incirca il 50% di probabilità di proseguire avendo un totale di 6 punti, eventualità che l'elevazione al cubo fa rapidamente aumentare con punteggi via via sempre minori. Il 441 che influisce sulla scelta del numero casuale non è altro che la somma dei cubi delle cifre dall'1 al 6

$1 + 8 + 27 + 64 + 125 + 216 = 441$

Anche se per il calcolo dei due punteggi si ripetono praticamente le stesse istruzioni, ecco spiegato il motivo principale per cui non si è usata una subroutine.

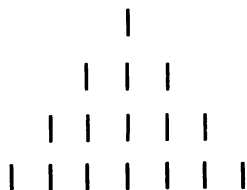
È infatti evidente che per il computer la comparsa della "matta" (L=9 e P=2 nelle linee 7 e 91) può avere un diverso significato che per te. Mentre tu puoi decidere se ti conviene o meno chiamare un'altra carta, per lui, determinando un possesso di almeno 7 punti, provocherà la fine della manche.

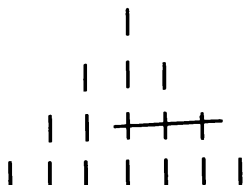
Interessante è l'attribuzione del punteggio relativo alla tanto attesa donna di cuori. Tenendo conto che mezzo punto viene assegnato in quanto essa è una figura, la somma totale sarà 7 se dopo tale assegnazione X (o Y)è costituito da un numero decimale, 7 e $^1/_2$ in caso di valore intero. Ciò si ottiene tramite la formula:

X (o Y) = 7.5 - FRAC X (o Y)

Non ce ne voglia l'insegnante di religione, ma chissà quante volte durante le sue ore, magari nel naturale proseguimento dell'intervallo, avrai dato vita con questo gioco ad appassionanti sfide con il tuo vicino di banco! Magari non conoscendo il nome di questo noto passatempo, che in effetti viene tramandato sotto diverse forme e fa uso di monetine, fiammiferi o altro, mentre nella versione "scolastica" la pratica consiste nel barrare delle aste precedentemente disegnate su un foglio.

Hai capito di che cosa si tratta? Eccoti quindi la trasposizione computerizzata, come al solito ben poco malleabile; ogni minimo errore ti costerà la partita. Comunque non tutto il male vien per nuocere: dopo un po' di allenamento contro il tuo Casio il compagno di banco ti sembrerà un dilettante al confronto, e sarai così in grado di superarlo molto più agevolmente.

....COME SI GIOCA

Nella versione da noi conosciuta il gioco del Nim consisteva nel disporre, su quattro file, rispettivamente 1,3,5 e 7 aste

e quindi cancellarne alternativamente, sbarrandole, quante si ritiene opportuno, purchè situate sulla medesima riga.

La prima mossa potrebbe ad esempio essere

Viene considerato vincente colui che riuscirà a barrare l'ultima asta ancora in gioco.

Troppo semplice? Fai qualche partita e poi ce lo sapremo ridire.

Qui, per evitare ripetizioni di schemi, abbiamo variato leggermente il "quadro" iniziale, consentendo una completa casualità nello stabilire quante aste formano ciascuna riga, numero che può variare da 0 a 9; potrà così talvolta capitare che vi sia in effetti un numero di righe minore di quattro.

Il programma prevede inoltre la possibilità di decidere chi effettua la prima mossa, il che avviene rispondendo alla domanda

COMINCI TU?

che ovviamente compare prima dello schema iniziale.

Ogni risposta diversa da "NO" lascerà a te l'onere, spesso determinante per il risultato finale, di dare il via alla contesa.

Essendo il display del Casio formato da una sola linea non si sono potute visualizzare effettivamente le quattro righe del gioco ma vengono riportati, sulla destra di un rombo (♦), i numeri delle aste presenti in ognuna di esse. Così

indica che nella 1ª riga vi sono 2 aste
nella 2ª riga vi sono 0 aste
nella 3ª riga vi sono 8 aste
nella 4ª riga vi sono 3 aste

Non appena compare il quadro, se è il tuo turno di gioco puoi effettuare la scelta, che apparirà all'estrema sinistra del display. Dovrai indicare dapprima la linea, e quindi il numero di aste che intendi togliere. Stai attento a non sbagliarti, perchè non è più possibile correggere gli errori di impostazione, anche se comunque il computer evita mosse "impossibili", impedendoti ad esempio, nello schema precedente, di considerare la seconda riga o di togliere 3 aste dalla prima.

Ricapitolando, se desideri ad esempio togliere 5 aste dalla terza linea dovrai digitare in successione 3 e poi 5 (senza premere EXE). Ti apparirà

3	5		♦		2		0		8		3

e poi il risultato dell'operazione

Successivamente il Casio effettuerà la sua mossa, fornendoti quasi istantaneamente il quadro che ne risulta

Se dovesse attendere un po' più del solito (comunque al massimo qualche secondo) è molto probabile che sia in difficoltà: cerca di approfittarne perchè è una eventualità non certo frequente. Se ti trovi perso non tentare di fare il furbo cercando di togliere zero aste da una linea: la tua mossa non verrà accettata.

Sei pronto a mettere alla prova la tua abilità logico-deduttiva? Non resta allora che augurarti buona fortuna, anche se in effetti qui la dea bendata non c'entra per nulla: la vittoria o la sconfitta dipendono solo da te.

IL LISTATO

```
FUORI PROGRAMMA IN MODO RUN:
Q=0   R=1   S=10   T=11   U=100   V=101
W=110   X=111   Y=1000   Z=1001   $="HAI VINTO TU HO VINTO IO!"
1     PRINT CSR 4; "NIM"
2     FOR I=1 TO 4: A(I)=INT(RAN#*10): NEXT I: INPUT "COMINCI TU",
      A$
3     IF A$="NO" THEN 14
4     GOSUB 49: J=0: GOSUB 40
5     IF K>0; IF K<5; IF A(K)>0; A=K: GOTO 7
6     GOTO 4
7     J=1: GOSUB 40: IF K≤A(A); IF K>0 THEN 12
```

```
9     PRINT CSR 1; " ƀ";: GOTO 7
12    A(A)=A(A)-K: K=1
14    GOSUB 49: FOR L=1 TO 4: FOR I=1 TO 4: L(I)=A(I): NEXT I
15    FOR I=1 TO A(L)
16    L(L)=A(L)-I: IF L(L)< 0 THEN 19
17    GOSUB 60: IF F=0; A(L)=L(L): GOTO 21
18    NEXT I
19    NEXT L: GOSUB 50
20    A(J)=INT(RAN#*G)
21    K=14: GOTO 4
40    K$=KEY: IF K$="" THEN 40
41    PRINT CSR J; K$;: K=VAL(K$)
42    FOR I=1 TO 99: NEXT I: RETURN
49    FOR I=1 TO 4: PRINT CSR 2*I+2; A(I); CSR 0; "ƀƀƀ◆";: NEXT I
50    G=0: FOR H=1 TO 4
51    IF A(H)>G; G=A(H): J=H
52    NEXT H: IF G≠0; RETURN
53    GOSUB 42: PRINT: PRINT MID(K,12): GOTO 2
60    H=0: F=0: FOR K=1 TO 4: H=H+Q(L(K)): NEXT K
61    FOR K=0 TO 3: A=INT(H/10↑K): IF A=0 THEN 63
62    IF A/2≠INT(A/2); F=1: RETURN
63    NEXT K: RETURN
```

Il segno ◆ nella riga 49 è fatto con MODE . SHIFT K
(resta 1 passo)

SE VUOI IMPARARE

PRINCIPALI VARIABILI

A(1)...A(4)	numero di aste nelle 4 righe
F	indica (0 od 1) la sicurezza o meno della mossa del Casio
G	max numero di aste presenti in una riga
H	somma delle conversioni in binario del numero delle aste presenti in ogni riga
J	indica alternativamente lo spazio di stampa (0 od 1) della tua scelta, ed in quale linea si trova G
K	cifra da te immessa (linea o n° di aste) ed in seguito indica quale parte di $ verrà visualizzata
L(1)...L(4)	copia del vettore A(1)...A(4)
Q...Z	conversione in binario delle cifre 0...9

Prima di addentrarci nella spiegazione del listato, per una sua completa comprensione è necessario chiarire qual è il procedimento adottato dal Casio per effettuare la sua mossa. Si tratta di una geniale strategia ormai molto conosciuta e scoperta dal Prof. Bouton dell'Università di Harvard che permette al computer di approfittare di ogni minimo errore dell'avversario e di essere quindi quasi imbattibile, in particolar modo se gli consentiamo di iniziare la partita. Tieni però presente che compiere la prima mossa rappresenta un effettivo vantaggio soltanto se hai ormai acquisito una buona dimestichezza con il gioco o, meglio, se anche tu conosci la strategia vincente. Vediamo di cosa si tratta.

Ogni configurazione di aste può, dopo ogni mossa, essere considerata "sicura" o "pericolosa" per chi l'ha giocata: sarà definita "sicura" una configurazione tale che, giocando correttamente, conduca alla vittoria; in caso contrario si tratterà ovviamente di una posizione "pericolosa".

Per vincere al NIM sarà quindi sufficiente fare in modo che, dopo ogni mossa, rimanga sul "tavolo" una posizione sicura. Tutto qui! Il problema è quindi essere capaci di stabilire la qualità di una configurazione.

Bouton suggerisce di trasformare il numero di aste di ogni riga in binario, incolonnare i risultati e sommare le cifre di ogni colonna: soltanto se tutte queste somme parziali sono pari o uguali a zero la posizione è sicura. Troppo complicato? Vediamo un esempio concreto:
la posizione 3 4 5 è insicura perchè

3 = 011 in base 2   4 = 100 in base 2   5 = 101 in base 2

e

$$\begin{array}{r} 011 \\ 100 \\ \underline{101} \\ 212 \end{array}$$

ed in effetti la somma delle cifre della colonna centrale è uguale ad 1, che è dispari.

Se hai compreso il meccanismo possiamo a questo punto addentrarci nel programma vero e proprio che è ovviamente centrato sull'algoritmo appena visto.

Dopo la memorizzazione nel vettore A(I) delle aste di ogni linea e dopo i necessari controlli relativi alla tua mossa, che viene eseguita nella linea 12, il gioco passa nelle "mani" del computer.
Come vedi (linea 14), vi è subito un richiamo alla subroutine 49, che svolge due funzioni: visualizza il nuovo quadro di gioco dopo ogni mossa (nella stessa linea 49) e ricerca il numero massimo (G) di aste contenute nel tabellone ed in quale linea (J) tale massimo è situato.

Lo scopo di questa ricerca è duplice, poichè indica innanzitutto se il gioco è

finito o se può continuare (IF G≠0; RETURN) ed inoltre sarà utile nel caso il tuo calcolatore non sia in grado, una volta tanto, di effettuare una mossa vincente, come vedremo fra breve.

Al ritorno dalla subroutine si trovano le istruzioni necessarie a far eseguire la mossa al tuo Casio, compito svolto mediante la tecnica seguente: dapprima le variabili A(I) vengono salvate nelle corrispondenti L(I) per evitare che vada perduto il loro valore originario, e quindi (ciclo su I nelle linee 15/18) vengono tolte una alla volta le aste dalle righe, facendo uso delle L(I).

Con la subroutine 60 si verifica la sicurezza o meno della mossa: in caso positivo (F=0) la mossa viene eseguita (A(L)=L(L)), altrimenti si prosegue nella ricerca della configurazione vincente.

Nel caso tutti gli sforzi risultassero vani la sua mossa consisterà nel togliere un numero casuale di aste dalla riga che ne ha di più (linea 20) il che chiarisce definitivamente l'utilità della ricerca di G vista in precedenza.

Non ti ha molto soddisfatto il metodo adottato dal Casio per giocare? Bello sforzo! - avrai certo pensato - Le prova tutte fino a trovare la mossa giusta. In effetti è così, ma non devi scandalizzarti troppo. L'unico grande vantaggio, almeno fino ad ora, che il computer possiede rispetto all'uomo è la grandissima rapidità di calcolo, che qui è adeguatamente sfruttata.

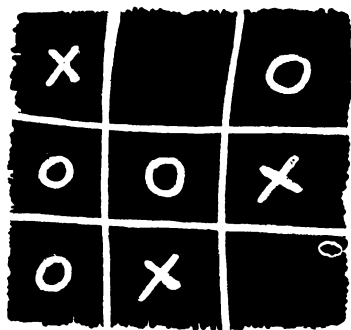
Se dovessimo provare a mano tutte le possibili combinazioni occorrerebbe molto tempo, alcune volte troppo per un gioco sufficientemente immediato, mentre un computer è in grado in pochi minuti di provare una grande quantità di soluzioni (pensa ad esempio agli scacchi, alla dama, ecc.).

Ma torniamo al nostro programma. Per completare la spiegazione manca ormai un solo tassello, la subroutine 60, relativa alla sicurezza della mossa.

La limitata quantità di memoria disponibile non ci ha permesso di trasformare all'occorrenza, all'interno del programma, il numero delle aste in binario. Si è potuto però ovviare inserendo in modo RUN, nelle variabili da Q a Z, la relativa conversione delle cifre da 0 a 9.

La verifica della parità di tutte le somme di ogni colonna è effettuata come segue: nel ciclo su K della linea 60 si esegue la somma ordinaria dei numeri binari, come se fossero in base 10; quindi (linea 61) il totale trovato viene diviso per potenze crescenti di 10 (1, 10, 100, 1000) considerando solo la parte intera. Se ogni parte intera è pari (linea 62) si ritorna al programma principale con F=0, altrimenti F viene posto uguale ad 1 indicante una configurazione pericolosa.

Hai capito tutto? Speriamo di si, anche perchè non è semplicissimo districarsi fra i vari L, L(L), A(L), A(A) presenti nel listato. Una breve nota merita inoltre la variabile K che, dopo essere usata per l'introduzione della tua mossa (subroutine 40) indicherà (linee 12 e 21) quale metà di $ deve essere stampata al termine del gioco (K=1 o K=14 nella linea 53).

IL GIOCO DEL 15

Un altro universale passatempo è costituito dal tic-tac-toe che tu forse conoscerai con il nome di Filetto. È un gioco per due persone che, alternativamente, collocano il proprio simbolo (0 o X) in una tabella 3 x 3

X		0
	X	
X	0	0

Vince chi riesce a disporre tre dei simboli che lo contraddistinguono in una qualsiasi fila, orizzontale, verticale o diagonale.

Non diciamo di più perchè sarebbe senz'altro superfluo, ma piuttosto rispondi ad una nostra domanda. Ti stai chiedendo quale nesso vi sia tra il filetto ed il gioco che ti stiamo presentando?

Una relazione esiste, ed anzi ben di più: in pratica si tratta della stessa cosa.

....COME SI GIOCA

Questa volta non sarai chiamato a combattere contro il computer, ma dovrai procurarti un avversario in carne ed ossa.

Entrambi, prima di iniziare la contesa, dovrete inserire i rispettivi nomi, facendo attenzione che non siano più lunghi di 7 caratteri in quanto, per evidente esaurimento dei passi disponibili, non ci è stato possibile includere alcuni comodi controlli.

Compiuta questa propedeutica operazione, che consentirà in seguito al tuo Casio di proclamare il vincitore, sul display compariranno le cifre dall'1 al 9 ed un cursore lampeggiante (-) posto in corrispondenza del primo carattere.

A questo punto tutto è pronto per iniziare e, cosa importante, dovrà farlo colui che ha inserito per primo il proprio nome, pena l'inversione del risultato finale.

Si tratta di selezionare, alternativamente, una delle cifre fra quelle rimaste disponibili, cercando di fare in modo che l'addizione di 3 fra quelle in nostro possesso dia come somma 15.

Per scegliere la cifra voluta dovrete dapprima spostare il cursore lungo il display usando i tasti

D → DESTRA
S → SINISTRA

fino a posizionarlo sul carattere prescelto, e quindi premere "X" per la conferma.

Supponiamo ad esempio che come prima mossa venga deciso di giocare il "4". Premendo D posizioneremo il cursore

1	2	3	4	5	6	7	8	9			

quindi pigiando la X vedremo apparire la nuova situazione, con il 4 che è stato prelevato dalla sequenza delle cifre giocabili e trasferito sulla destra del quadro, tra due simboli "■"

1	2	3	5	6	7	8	9	■	4	■	

Ovviamente la possibilità di scelta del secondo giocatore è ristretta ad otto cifre, e quella da lui indicata verrà posta alla destra del display, ad esempio

1	2	3	6	7	8	9	■	4	■	5	■

Dopo ogni mossa la situazione cambierà, restando però immutato l'ordine delle tre sequenze: le cifre ancora "libere", poi quelle in possesso del primo giocatore ed infine le disponibilità del suo avversario.

Il gioco proseguirà sino a quando uno dei due non totalizzerà 15 come somma di tre delle cifre da lui scelte o, senza vincitori né vinti, nell'ipotesi di esaurimento dei numeri.

Per concedere la rivincita sarà sufficiente, come al solito, soltanto la pressione di EXE mentre, nel caso ad esempio di un piccolo torneo, se cambiano i concorrenti dovrai dare nuovamente il RUN.

Hai capito a questo punto la strettissima relazione con il gioco del filetto? Noo? Vediamola insieme.

Le possibili terne di numeri che forniscono come somma 15 sono 8 in tutto

1 5 9   1 6 8   2 4 9   2 5 8

2 6 7   3 4 8   3 5 7   4 5 6

e, se ci pensi bene, è possibile disporle in una scacchiera 3 x 3 in modo che ogni linea orizzontale, verticale o diagonale dia appunto la fatidica somma di 15, ad esempio

8	1	6
3	5	7
4	9	2

Si forma così un cosiddetto quadrato magico di lato 3, in cui ogni terna vincente corrisponde ad un "filetto".

Eccoti quindi su un piatto d'argento un trucchetto, forse un po' sleale, per sbaragliare il tuo avversario: sarà sufficiente memorizzare il quadrato e, sapendo giocare razionalmente il filetto, potrai fare altrettanto con questa sua trasposizione.

IL LISTATO

```
2    PRINT "GIOCO DEL 15": FOR I=1 TO 9 STEP 8: INPUT "NOME",
     A$(I): NEXT I
5    $="123456789": C=0: K=0: Q=9
6    FOR I=1 TO 9 STEP 8: A=0
7    PRINT CSR A; "-"; CSR 0; $;: IF KEY="D"; IF A<Q-1; A=A+1
9    IF KEY="S"; IF A>0; A=A-1
10   IF KEY≠"X" THEN 7
11   T$=MID(A+1,1): B(I)=B(I)+1: U=B(I): IF Q=1; $="": GOTO 13
12   $="ƀ"+$+"ƀ": $=MID(1,A+1)+MID(A+3,Q-A): $=MID(2,Q-1)
13   B$(I+U)=T$: PRINT: PRINT $; "■";: FOR R=1 TO 9 STEP 8
14   IF B(R)≠0; FOR S=1 TO B(R): PRINT B$(R+S);: NEXT S: PRINT
     "■";
15   NEXT R: Q=LEN($): IF U<3 THEN 23
16   Y=0: FOR W=1 TO U: V$=B$(W+I): Y=Y+VAL(V$): NEXT W
17   IF U=3; Z=Y: GOSUB 50: GOTO 23
18   FOR W=1 TO 4: V$=B$(W+I): X=Y-VAL(V$): IF U=4; Z=X:
     GOSUB 50: GOTO 22
20   FOR R=W+1 TO 5: V$=B$(R+I): Z=X-VAL(V$): GOSUB 50:
     NEXT R
22   NEXT W
23   IF Q≠0; NEXT I: GOTO 6
24   PRINT: PRINT "PAREGGIO": GOTO 5
50   IF Z≠15; RETURN
52   PRINT: PRINT "VINCE:"; A$(I): GOTO 5
```

Il segno ■ nelle righe 13 e 14 è fatto con MODE . SHIFT Z
(resta 1 passo)

SE VUOI IMPARARE

PRINCIPALI VARIABILI

A	posizione del cursore
B$,J$	nomi dei giocatori
C,K	mosse effettuate dai due giocatori
B$(2)...B$(6)	cifre possedute dal 1° giocatore
B$(10)..B$(14)	cifre possedute dal 2° giocatore
Q	n° di scelte ancora possibili
T$	cifra selezionata
U	mosse effettuate dal giocatore di turno
V$	copia temporanea di un elemento di B$()
Y	somma delle cifre possedute dal giocatore di turno

Ecco un classico esempio, almeno secondo noi, di un gioco apparentemente semplice il cui listato non risulta forse altrettanto facilmente comprensibile. Almeno in parte ciò è determinato dalla consueta necessità di scrivere linee multistatement ma qui si aggiunge senz'altro una organizzazione dei dati ed una gestione delle variabili che ci sembra opportuno chiarire.

Per poter inserire tutto in un ciclo, in modo che la distinzione fra quello che si riferisce all'uno o all'altro dei due giocatori venisse fatta automaticamente, sono stati adoperati alcuni vettori, che vedremo in dettaglio.

Stabilito che la variabile A conserva la posizione del cursore, le lettere B...H sono utilizzate per memorizzare i dati che si riferiscono a chi inizia per primo, quelle J...P per il secondo.

A$(1)	= B$	NOME	J$= A$(9)
B(1)	= C	n° delle mosse effettuate	K= B(9)
B$(2)	= D$	1ª cifra posseduta	L$= B$(10)
B$(3)	= E$	2ª cifra posseduta	M$= B$(11)
B$(4)	= F$	3ª cifra posseduta	N$= B$(12)
B$(5)	= G$	4ª cifra posseduta	O$= B$(13)
B$(6)	= H$	5ª cifra posseduta	P$= B$(14).

Dovrebbe quindi essere chiaro che, essendo i cicli su R e su I che incontrerai nel listato eseguiti da 1 a 9 con STEP di 8, A$(I) seleziona il nome, B(I) o B(R) indicano il numero delle mosse effettuate, mentre quando compare B$() si tratterà di una delle cifre prelevate da uno dei contendenti. Le possibilità di scelta sono in tutto 9, ed effettuando un turno ciascuno è logico che le "prese" di un giocatore possono essere al massimo cinque anche se la partita può concludersi prima che tale limite venga raggiunto. Ma passiamo ora all'algoritmo vero e proprio.

L'alternanza di mosse è svolta con il ciclo su I della linea 6 che occupa praticamente tutto il listato (fino alla linea 23).

Il movimento del cursore (linee 7 e 9) avviene nel solito modo tenendo conto in questo caso che il campo di variabilità diminuisce man mano che le cifre vengono prelevate da $, la cui lunghezza Q determina quindi il limite destro.

Dopo aver operato la scelta tramite la pressione del tasto "X", nella linea 11 viene incrementato il contatore delle mosse B(I) relativo al giocatore di turno (poi assegnato ad U per evitare complicazioni di notazione) e in seguito $ viene adeguatamente modificato.

Nella 13 il numero prelevato T$ viene attribuito a chi di dovere mediante l'istruzione B$(I+U)=T$. Forse la scrittura è un po' complicata ma un esempio ti dovrebbe chiarire tutto.

Se il 1° giocatore (I=1) ha appena eseguito la terza mossa (U=3) risulterà che B$ (I+U)=B$(4) che, come risulta nello schema dato all'inizio del paragrafo, corrisponde proprio alla terza cifra posseduta dal primo contendente.

Il ciclo su R che inizia alla linea 13 svolge il compito di visualizzare, dopo le cifre ancora disponibili ($), quelle possedute dai due avversari sempre che, naturalmente, essi ne abbiano già prelevato qualcuna (IF B(R) ≠ 0).

Subito dopo inizia la fase forse più complicata del programma: il controllo per stabilire se è stata ottenuta o meno una configurazione vincente. Sembrerà strano, ma mentre per un "essere umano" è molto semplice, con una rapida occhiata, stabilire se tre delle cifre possedute forniscono come somma 15, per quanto riguarda il computer abbiamo faticato non poco per compattare il tutto in poche istruzioni, senza far eseguire una serie interminabile di IF...THEN che, oltretutto, avrebbe occupato molti preziosissimi passi di memoria.

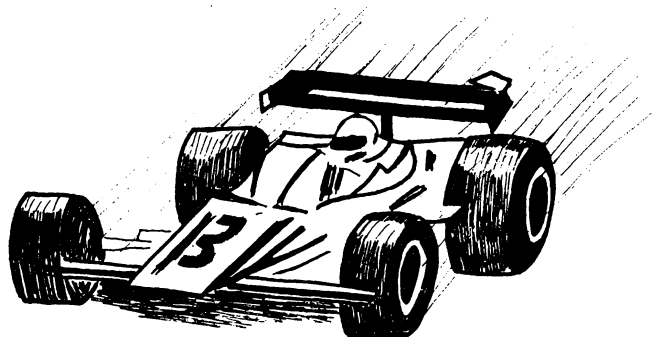
Senza la pretesa di aver trovato la soluzione ottimale, vediamo comunque come è stato risolto il problema.

Una delle difficoltà era costituita dal non sapere a priori su quante cifre doveva essere eseguito il controllo, potendo essere 3,4 o 5. Ovviamente possedendone meno di 3 non c'è nulla da verificare e la mano sarà ceduta direttamente all'avversario (IF U<3 THEN 23 nella linea 15), mentre in caso contrario non abbiamo trovato di meglio che distinguere le tre possibilità.

Innanzitutto (linea 16) viene calcolata la somma di tutte le cifre di un giocatore (Y) e quindi se U era uguale a 3 si verifica soltanto se Y è pari a 15 (subroutine 50).

Se le cifre possedute da un giocatore sono più di 3, con i due cicli annidati su W e su R nelle linee 18/22, vengono considerate tutte le somme parziali, non tenendo conto ora della 1ª, ora della 2ª, ora della 3ª ed infine della 4ª cifra posseduta (ogni volta assegnata a V$).

Ognuna di queste somme parziali sarà memorizzata nella variabile X e così se U era pari a 4 sarà sufficiente andare a controllare alla solita subroutine 50 se in qualche caso X è uguale a 15. Se invece siamo al termine del gioco, cioè se è stato prelevato anche l'ultimo numero disponibile, U sarà uguale a 5, ed è necessario (linea 20) togliere ancora da X il valore di un'altra cifra fra quelle rimaste, in modo che vengano prese in considerazione tutte le possibili combinazioni di tre "prese".

Ovviamente se anche con l'ultima mossa (Q=0) un giocatore non è riuscito a superare l'avversario la gara terminerà con un equo "PAREGGIO".

Se hai un debole per la Formula uno, se qualche volta hai sognato di essere Lauda o Patrese, Piquet o Arnoux questo gioco non mancherà di coinvolgerti.

Ora sei tu l'abile, spericolato e forse un po' pazzo pilota che ama sfrecciare nella pista a velocità incredibili, a bordo di una eccezionale vettura. Il circuito non è facile e le curve sono molto pericolose, ma oggi è il tuo grande giorno.

Al segnale di START lanciati a razzo e la pole position sarà tua.

....COME SI GIOCA

Abbiamo cercato in questo gioco di rappresentare la corsa di un'automobile lungo un circuito, cosa non semplice considerate le capacità grafiche del CASIO, ottime per un pocket ma pur sempre limitate.

A noi sembra di essere riusciti discretamente nell'intento, anche se è ovvio che il nostro "Circuito" non può competere con altre versioni più sofisticate presenti in commercio ed anche nelle sale - giochi. Ma passiamo alle spiegazioni.

L'automobile è simbolizzata dal segno "#" e tu dovrai guidarla nel circuito facendo attenzione a non mandarla fuori pista. Per le manovre dovrai destreggiarti tra ben nove tasti.

Questo è lo schema di guida:

TASTO	EFFETTO
F	dà inizio alla corsa (vel. 0)
D	l'auto si sposta a sinistra
G	l'auto si sposta a destra
R	l'auto frena e va sempre dritta
V	l'auto accelera e va sempre dritta
E	frena e contemporaneamente si sposta verso sinistra
C	accelera e contemporaneamente si sposta verso sinistra
T	frena e contemporaneamente si sposta verso destra
B	accelera e contemporaneamente si sposta verso destra

Riassumendo si ha:

E	R	T
D	F	G
C	V	B

TASTI

F/S	F	F/D
S	START	D
A/S	A	A/D

EFFETTO

F = frena
A = accelera
D = destra
S = sinistra

Ti vediamo un po' perplesso, ma davvero non è il caso. Anche se i tasti del gioco sono molti non è difficile usarli correttamente, una volta che avrai ben imparato la tabella e ancor meglio dopo aver verificato la loro specifica funzione.

Per tutto il corso della prova a destra del display, vedrai visualizzati una lettera ed un numero che ti forniscono utilissime indicazioni.

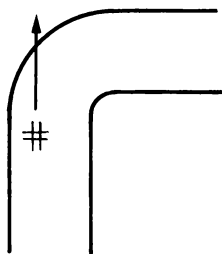
La lettera si riferisce all'andamento ed alla struttura del circuito, per cui all'apparire di "R" saprai che sei in rettilineo; "D" indica una curva a destra ed "S" una curva a sinistra

R→RETTILINEO D→DESTRA S→SINISTRA

Il numero rappresenta invece la velocità dell'automobile (da 0 a 9).

Come vedi, niente di difficile e a questo punto puoi partire.

Per iniziare la corsa è necessario premere il tasto F di start e quindi accelerare subito tramite V, per non mantenere l'automobile a velocità 0; nel qual caso il gioco non terminerebbe mai.

Sono indispensabili ottimi riflessi per spostarsi rapidamente a destra o sinistra all'apparire delle curve ed anche una buona tecnica di guida che consentirà di accelerare e frenere al momento opportuno. Se esiterai anche solo di un poco a portare la tua auto ad esempio a destra all'apparire della "D", essa si verrà a trovare immediatamente all'estremità sinistra della curva e quindi andrà FUORI PISTA.

Ai primi tentativi tale inconveniente è praticamente inevitabile, ma una volta che avrai imparato il circuito (che è sempre lo stesso) tutto sarà più semplice in quanto saprai in anticipo quale direzione far prendere alla tua automobile; ciò ti risulterà utile soprattutto in prossimità delle curve.

È consigliabile cercare dapprima di compiere l'intera pista a velocità ridotta (1 o 2) e solo in un secondo tempo gareggiare sempre più velocemente, per ottenere ottimi tempi di percorrenza. Infine puoi sempre operare sulla linea 2 del listato e modificare il circuito a tuo piacimento per renderlo, se lo desideri, sempre più complicato e tortuoso.

IL LISTATO

```
 1   VAC
 2   $="9R3R8D9R3R7S5D4R4D3S6D7R9D4S9R": A=4: GOSUB 30
 3   IF KEY≠"F" THEN 3
 4   FOR L=1 TO 29 STEP 2: B$=MID(L,1): X=VAL(B$)
 6   IF KEY="" THEN 18
 7   IF V=0 THEN 12
 8   IF KEY="E"; V=V-1: A=A-1: GOTO 17
 9   IF KEY="T"; V=V-1: A=A+1: GOTO 17
10   IF KEY="R"; V=V-1: GOTO 17
11   IF V=9 THEN 15
```

```
12    IF KEY="C"; V=V+1: A=A-1: GOTO 17
13    IF KEY="B"; V=V+1: A=A+1: GOTO 17
14    IF KEY="V"; V=V+1: GOTO 17
15    IF KEY="D"; A=A-1: GOTO 17
16    IF KEY="G"; A=A+1
17    PRINT CSR 10; V; CSR 10; Y$;
18    Y$=MID(L+1,1): K=0: IF Y$="D"; K=-1
19    IF Y$="S"; K=1
21    W=V-R: T=T+2: IF X≥W; X=X-W: A=A+W*K: GOSUB 30:
      R=0: GOTO 6
22    T=T+1: A=A+X*K: R=W-X: GOSUB 30: NEXT L
24    PRINT: PRINT "FINE T="; T/100: END
30    IF A≤0; A=0: H=1
31    IF A≥9; A=9: H=1
32    PRINT: PRINT "■"; CSR 9; "■"; CSR A; "#"; CSR 10; V; CSR 10; Y$;
33    IF H=0; RETURN
34    PRINT: PRINT "FUORI PISTA"
```

Il segno ■ nella riga 32 è fatto con MODE . SHIFT Z

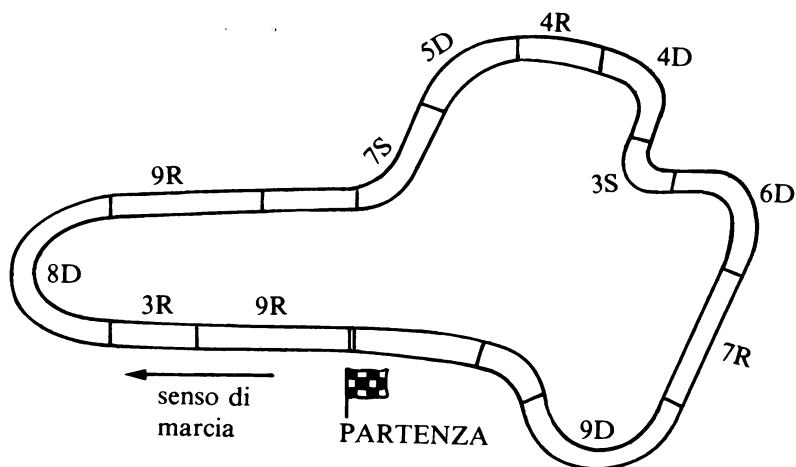
Il segno # nella riga 32 è fatto con SHIFT E

(resta 1 passo)

SE VUOI IMPARARE

PRINCIPALI VARIABILI	
A	spazio di stampa dell'automobile
T	tempo impiegato a percorrere la pista
V	velocità dell'automobile
X	lunghezza del tratto di pista da affrontare
Y$	è D, R o S a seconda dell'andamento del circuito
$	contiene il circuito

Abbiamo lasciato per ultimo questo gioco in quanto, per la realizzazione della corsa dell'automobile lungo il circuito, è stato necessario ideare ed elaborare un algoritmo a nostro avviso piuttosto "cervellotico" e certamente meno standard rispetto ai precedenti. Cercheremo di ripercorrere insieme le varie fasi che ci hanno portato alla stesura finale del programma.

Si doveva innanzitutto trovare un metodo adeguato per memorizzare il circuito preliminarmente disegnato sulla carta.

La soluzione migliore da noi escogitata è stata quella di suddividere la pista in diversi segmenti a seconda della direzione (destra, sinistra o rettilineo) e di riportarne le lunghezze approssimative nella variabile $, l'unica capace di contenere un numero piuttosto elevato di caratteri.

In tale stringa le misure dei vari tratti sono seguite dall'iniziale che ne indica il senso di percorrenza (R - D - S).

L'avanzare del tuo bolide nel circuito avviene all'interno del ciclo su L, dove di volta in volta vengono estratte da $ le caratteristiche del tratto che devi affrontare. Ad esempio all'inizio verranno selezionati i caratteri 9R indicanti, come già detto, un RETTILINEO lungo 9.

Ovviamente non si fa riferimento ad alcuna unità di misura ma, se ti risulta più semplice, puoi pensare che corrispondono a 9 Km.

Il tempo per cui resterai in un determinato tratto è in relazione sia alla sua lunghezza che alla velocità impressa alla tua auto. Dovendo ad esempio percorrere una parte di circuito lunga 9 alla velocità 2, ti occorreranno 5 unità di tempo per completarla e passare così a quella seguente.

Dopo la prima unità di tempo (data per noi dal ritardo tra due successivi calcoli relativi alla posizione della vettura) hai infatti percorso i 2/9 della distanza, poi i 4/9 e così via, se naturalmente mantieni sempre la medesima velocità 2.

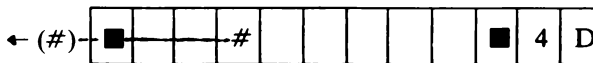
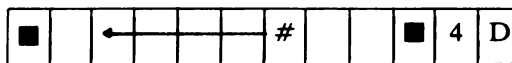
Ogni volta il tratto di strada ancora da affrontare diminuirà di conseguenza (9 - 7 - 5 - 3 - 1) e la R che indica il rettilineo continuerà ad essere estratta e a comparire nel display fino a che i ripetuti decrementi non azzereranno la lunghezza iniziale.

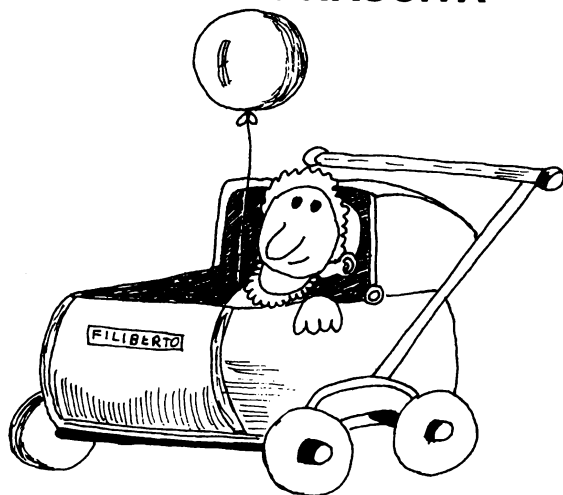
Come nell'esempio fatto può accadere che lo 0 non sia raggiungibile esattamente; in questo caso rimarrà un resto (R) che influenzerà i calcoli successivi.

Nella linea 21 infatti per ottenere la nuova posizione dell'auto (A) non viene usata l'effettiva velocità (V) ma la sua correzione W = V - R.

Vediamo ora come, oltre ai comandi di guida da te impostati, il movimento del bolide è legato all'andamento della pista.

È evidente che dovendo percorrere una curva si è in presenza di una forza centrifuga che tenderà a portarti dalla parte opposta rispetto al tuo tentativo di sterzata; per rendere questa situazione è stato attribuito alla variabile K un diverso valore (-1, 0, 1) a seconda della lettera estratta (D, R, S).

Così in una curva a destra K varrà -1, e se possediamo ad esempio una velocità di 4, mediante il calcolo A = A + W * K si otterrà un decremento di A pari a 4, corrispondente ad uno spostamento tendenziale verso sinistra di ben 4 posizioni.

Se l'auto fosse dove indicato in un attimo ti ritroveresti fuori pista; soltanto una velocità diminuita in tempo o un diverso approccio alla curva (più sulla destra)

avrebbero potuto salvarti.

GIORNO DI NASCITA

Ti sarai forse domandato perchè abbiamo definito il precedente "Circuito" come ultimo gioco, mentre in effetti ne mancano ancora due all'appello. Beh, è presto detto! Gli ultimi due listati che ti presentiamo più che veri e propri giochi di competizione sono dei semplici divertimenti nei quali tu, dopo aver fornito al Casio i dati iniziali, non dovrai far altro che attendere la sua risposta.

Niente più smanettamenti quindi, nè l'uso delle tue capacità strategiche o dei tuoi riflessi; siamo comunque certi che anche questi ultimi due programmi verranno eseguiti moltissime volte, come ci auguriamo sia accaduto per i precedenti. Vediamo di cosa si tratta.

Sembrerà strano ma, 9 volte su 10, una persona non ricorda mai in quale giorno della settimana è nato e per ovviare a questa lacuna ecco pronta la soluzione.

Ti verranno richiesti tre dati: innanzitutto dovrai inserire la lettera (M o F) che indica il tuo sesso, poi il nome ed infine la tua data di nascita.

Bada bene che quest'ultima dovrà essere digitata in un unico INPUT, nella forma GIORNO MESE ANNO, con i rispettivi numeri separati da uno spazio, ad esempio

21 7 1964

Premendo EXE vedrai apparire la risposta al tuo interrogativo.

È ovvio che, oltre a sapere quando sono nati i tuoi familiari e tutti gli amici, il programma può essere usato anche per altri scopi.
Vuoi forse sapere in che giorno è morto Napoleone?

Non dovrai far altro che inserire

Sesso?	M
Come ti chiami?	Napoleone
DATA DI NASCITA?	5 5 1821

ed otterrai

NAPOLEO, SEI NATO DI SABATO

In questo caso il programma non è in grado di distinguere la nascita dalla morte, ma d'altra parte nessuno è perfetto.

Qualche breve nota al listato.

L'algoritmo usato (linee 360/410) deriva da cognizioni astronomiche che neanche noi possediamo, ed è quindi da prendere così com'è, fiduciosi della sua esattezza, del resto dimostrata dalle numerose prove effettuate.

L'unica cosa degna di nota è la subroutine 1000, con la quale la data di nascita immessa in $ viene scissa nei tre numeri (giorno, mese, anno) richiesti dall'algoritmo.

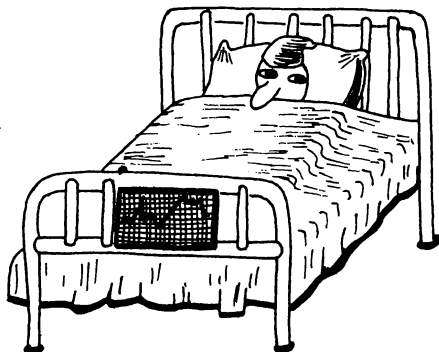
Un piccolo scherzetto quasi in conclusione del libro (eh!, eh!) sarà quello di non dirti più nulla, invitandoti a meditarci sopra, in particolare sulla linea 1010 (FOR K = K + 1 TO L).

IL LISTATO

```
 10  INPUT "Sesso M/F", H$
 15  Y$="NATO": IF H$="F"; Y$="NATA"
 20  INPUT "Come ti chiamiƀ", $
 30  IF LEN($)>7; W$=MID(1,7): GOTO 110
 40  W$=$
110  INPUT "DATA DI NASCITAƀ", $
120  L=LEN($): K=0
140  GOSUB 1000: C$=B$: GOSUB 1000: D$=B$: GOSUB 1000:
     E$=B$
200  G=VAL(C$): A=VAL(E$): M=VAL(D$)
360  IF M<3; M=M+12: A=A-1
370  S=G+2*M+INT(3*(M+1)/5)+A+INT(A/4)
380  IF A>1582; R=INT(A/100)-INT(A/400)-2: GOTO 400
```

```
 390  R=0
 400  T=S-R+5
 410  N=T-INT(T/7)*7+1
 415  PRINT W$; ",SEIƀ"; Y$; " ƀDIƀ";
 420  IF N=1; PRINT "LUNEDIƀƀ"
 430  IF N=2; PRINT "MARTEDIƀ"
 440  IF N=3; PRINT "MERCOLEDI"
 450  IF N=4; PRINT "GIOVEDIƀ"
 460  IF N=5; PRINT "VENERDIƀ"
 470  IF N=6; PRINT "SABATOƀƀ"
 480  IF N=7; PRINT "DOMENICA"
 490  END
1000  B$=""
1010  FOR K=K+1 TO L
1020  Z$=MID(K,1)
1030  IF Z$=" ƀ"; RETURN
1040  B$=B$+Z$
1050  NEXT K: RETURN
```

(resta 1 passo)

BIORITMO

Forse oggi non ti senti troppo bene: niente di definito, ma senza un preciso motivo sei giù di tono, abbattuto sia fisicamente che mentalmente.

Spesso in certi casi il capro espiatorio è il tempo, considerato la causa di tutti i nostri malanni, ma fai attenzione a non accusare un "innocente"; talvolta sarebbe meglio prendersela con il proprio bioritmo.

Certamente anche tu avrai sentito parlare almeno una volta di questa teoria, secondo la quale il corpo umano possiede degli orologi interni, o meglio determinati ritmi metabolici. Si pensa infatti che esistano tre cicli che hanno inizio dalla nascita in senso positivo e che oscillano sempre tra un valore minimo ed uno massimo ripetendosi periodicamente, cioè assumendo di nuovo un determinato valore dopo un prefissato numero di giorni.

Il ciclo FISICO (durata 23 giorni) riguarda la salute, la vitalità e l'energia; il ciclo EMOTIVO (durata 28 giorni) riguarda la sensibilità, la sfera affettiva, i rapporti con gli altri; il ciclo MENTALE (durata 33 giorni) ha come oggetto la prontezza e la capacità intellettuale.

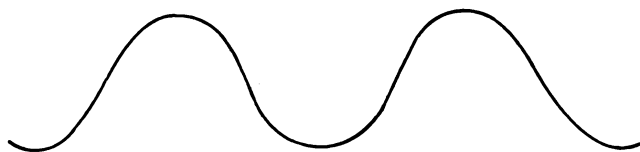
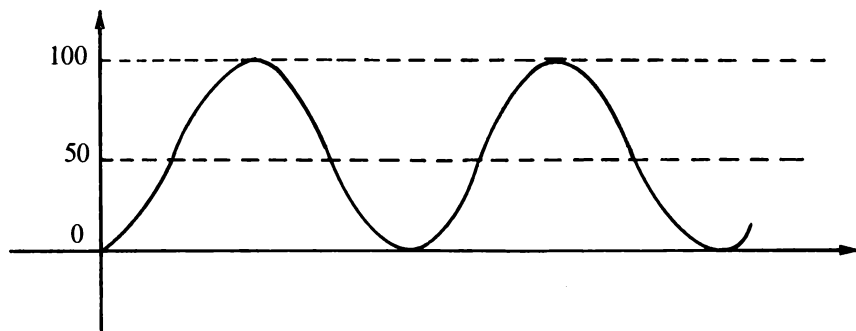
Anche se la sua scientificità non è ancora dimostrata c'è sicuramente chi considera questo argomento non un gioco ma una cosa seria e sempre più spesso è usato in diverse applicazioni. Pare ad esempio che i giapponesi

elaborino i bioritmi dei dirigenti per evitare errori di valutazione, senza parlare poi dell'ambiente sportivo in genere: sono sempre di più gli atleti che programmano la loro attività in relazione ai propri cicli metabolici.

Se anche tu appartieni al rilevante numero di persone interessate al bioritmo eccoti quindi la possibilità di calcolarlo, verificando di persona l'attendibilità o meno della teoria. Vedrai che, anche se non molto sofisticato, il programma che ti presentiamo sarà sicuramente divertente.

Dopo il titolo ti saranno richieste due date: con 3 differenti INPUT (al contrario di come avveniva nel gioco precedente) dovrai immettere dapprima quella di oggi, nella forma GIORNO, MESE, ANNO, ed in seguito la tua data di nascita.

Dopo qualche attimo di attesa ti verrà il responso dal computer: vediamo come valutarlo.

I tre cicli sopra menzionati hanno un cosiddetto andamento sinusoidale, cioè della forma

e noi abbiamo stabilito una scala di valutazione da 0 a 100

Quindi, per ogni ciclo presentato, il valore di 50 costituirà la via di mezzo tra un bioritmo positivo ed uno negativo e, proprio per questo suo significato di spartiacque, assume particolare importanza: la teoria ci informa infatti che i giorni più critici sono proprio quelli in cui le curve passano o sono prossime al 50, poichè in tal caso siamo in un periodo di transizione e più esposti ad eventuali pericoli.

Eccoti comunque qualche indicazione di massima

CICLO	da 0 a 48	49-50-51	da 52 a 100
FISICO	bisogno di riposo: ridurre le attività	incidenti in agguato	ottima salute, forza e resistenza fisca
EMOTIVO	tristezza malinconia	insta-bilità	relazioni sociali molto positive
MENTALE	evitare decisioni e responsa-bilità, rimandare l'interrogazione	errori di valuta-zione	grande senso del giudizio, ottime possibilità in campo scolastico

Per dare una completa informazione sull'andamento delle curve, oltre ad un numero compreso fra 0 e 100 si visualizzerà anche una freccia, verso l'alto (↑) o verso il basso (↓) a seconda che in quel giorno il ciclo sia ascendente o discendente.

Poichè i valori che compaiono sono interi potrebbe capitarti di vedere un 0↓ : ciò rappresenta ovviamente un'approssimazione e indica che per il raggiungimento del minimo è necessario percorrere ancora un piccolissimo tratto.

Ma eccoti ora, se vuoi saperne di più, qualche cenno sulle formule usate nel programma.

Nelle linee dalla 20 alla 50 si usa un complicato algoritmo, da prendere per buono, che calcola il numero di giorni trascorso da un ipotetico 1 gennaio dell'anno zero. È evidente che applicandolo per due volte (data di oggi e data di nascita), la differenza dei valori ottenuti, aumentata di 1, indicherà proprio quanti giorni (A nella linea 70) sono trascorsi dalla tua nascita.

Per il calcolo del bioritmo si è usata la formula

$$P = (\text{SIN} \frac{Z*360}{V} + 1) * 50$$

considerandone poi la sola parte intera.

In tale espressione

V = periodo del ciclo in giorni (23,28 o 33)

e

Z = A - V*INT A/V

cioè Z è uguale al resto della divisione fra il giorno del ciclo sotto esame (A) ed il suo periodo (V).

Ciò consente, come richiesto, di avere un argomento della funzione SIN sempre minore di 1440.

Infine per stabilire se il bioritmo è in ascesa o meno si confronta semplicemente il valore trovato W (non ancora esteso fra 0 e 100) con quello J del giorno successivo.

IL LISTATO
(DEFM 3)

```
  1   PRINT " ƀƀBIORITMO"
  5   FOR E=1 TO 2
 10   IF E=1; INPUT "data di oggiƀ", G(E),M(E),A(E): GOTO 20
 15   INPUT "data di nascita", G(E),M(E),A(E)
 20   S(E)=365*A(E)+G(E)+31*(M(E)-1)
 30   IF M(E)<3; A(E)=A(E)-1: GOTO 50
 40   S(E)=S(E)-INT((4*M(E)+23)/10)
 50   S(E)=S(E)+INT(A(E)/4)-INT(3*(1+INT(A(E)/100))/4)
 60   NEXT E
 70   A=T-U+1
 75   T=0
 80   FOR V=23 TO 33 STEP 5
 90   T=T+1
 94   J=INT(A/V): Z(T)=A-(J*V)
100   W(T)=SIN(Z(T)*360/V)
110   J(T)=SIN((Z(T)+1)*360/V)
120   P(T)=INT((W(T)+1)*50)
130   IF W(T)<J(T); D$(T)="↑": GOTO 150
140   D$(T)="↓"
150   NEXT V
160   PRINT "FISICOƀ"; Q; E$;: GOSUB 200
170   PRINT "EMOTIVO"; R; F$;: GOSUB 200
180   PRINT "MENTALE"; S; G$: END
200   FOR V=1 TO 500: NEXT V: PRINT: RETURN
```

Il segno ↓ nella linea 140 è fatto con MODE . SHIFT O

(restano 15 passi)

APPENDICE A

RICHIAMI SUL BASIC DEL PB-100

GESTIONE DELLA MEMORIA

Il computer è venduto con 544 Byte di memoria RAM (Random Access Memory) utilizzabili dall'utente ed espandibili fino a 1568. Questi byte possono essere suddivisi tra passi di programma e numero di variabili disponibili, tenendo conto che ogni variabile aggiuntiva rispetto alle normali 26 (le 26 lettere dell'alfabeto inglese) costa la soppressione di 8 passi di programma. La ripartizione voluta si effettua tramite il comando DEFM, dove viene indicato il numero di variabili aggiuntive che è possibile utilizzare. Ad esempio con DEFM 10 si hanno a disposizione 36 memorie e 464 passi di programma.

Di seguito riportiamo la tabella delle possibili situazioni

DEFM	NUMERO DI VARIABILI	PASSI DI PROGRAMMA DISPONIBILI	
		SENZA ESPANSIONE	CON ESPANSIONE
0	26	544	1568
1	27	536	1560
2	28	528	1552
3	29	520	1544
4	30	512	1536
:	:	:	:
10	36	464	1488
:	:	:	:
30	56	304	1328
:	:	:	:
40	66	224	1248
:	:	:	:
60	86	64	1088
:	:	:	:
67	93	8	1032
68	94	0	1024
69	95		1016
:	:		:
74	100		976
:	:		:
174	200		176
:	:		:
196	222		0

La memoria di programma è suddivisa in 10 zone (dalla 0 alla 9) nelle quali è possibile memorizzare fino a 10 differenti programmi, indipendenti gli uni dagli altri, potendo cioè utilizzare ad esempio gli stessi numeri di linea, con l'unica ed ovvia limitazione della quantità di memoria ancora disponibile.

Per selezionare la zona desiderata è sufficiente premere SHIFT seguito dalla cifra corrispondente

SHIFT 0 seleziona la zona 0

MODO RUN

All'accensione il computer si trova già in modo "RUN".
Se, essendo in modo "WRT", si desidera passare in modo "RUN" è sufficiente premere MODE 0.
In modo "RUN" è possibile eseguire calcoli manuali o far partire l'esecuzione di un programma.

Introducendo in successione A=100 (EXE)
B=50 (EXE)

i valori 100 e 50 verranno assegnati alle variabili A e B.
Se a questo punto si batte A+B (EXE) verrà visualizzato il valore 150 (somma dei valori contenuti in A e in B).
Quando si esegue un calcolo manuale, per ottenere il risultato finale occorre premere il tasto (EXE) e non quello di (=), essendo quest'ultimo usato per l'assegnazione di valori o di stringhe alle variabili.
Per comandare l'esecuzione di un programma, posto ad esempio nella zona 3, si può battere SHIFT 3, ed esso girerà automaticamente. Altra possibilità è quella di usare RUN (EXE) dopo aver selezionato la zona di memoria desiderata.
RUN 12 (EXE) farà partire l'esecuzione della linea 12
Se durante lo svolgimento di un programma viene trovato un errore l'elaborazione cessa e viene trasmesso un messaggio indicante il tipo di errore e la riga in cui è stato riscontrato (vedi Appendice C) e prima di poter effettuare le modifiche del caso è necessario premere "AC" per cancellare il display.

MODO WRT

Se, essendo in modo "RUN", si vuole passare in modo "WRT" è sufficiente premere MODE 1.
In modo "WRT" verranno scritti i programmi nei quali ciascuna linea deve essere numerata con un numero intero compreso fra 1 e 9999. Le varie righe possono essere introdotte in ordine sparso: provvederà il computer a disporle in ordine crescente.
In ogni linea si possono scrivere più istruzioni purchè separate dai due punti (:) e purchè non superino il limite massimo di 62 caratteri compreso il numero di linea stesso.

Ogni linea di programma occupa un certo numero di passi così calcolabili:
— 2 passi per il numero di linea
— 1 passo per la memorizzazione della linea (EXE)
— 1 passo per ogni comando o funzione (PRINT,SIN,NEXT,ecc.)
— 1 passo per ogni altro carattere

110 PRINT "CIAO" occupa 10 passi

Gli spazi, eccetto quelli racchiusi fra virgolette, non occupano passi di memoria.

Per cancellare la zona corrente si deve impostare, in modo "WRT", il comando CLEAR (EXE). Se si desidera cancellare tutte le zone di memoria si dovrà battere CLEAR A (EXE)

VARIABILI

Le variabili possono essere indicate con una lettera dell'alfabeto o essere considerate come elementi di un vettore (vedere in seguito il relativo paragrafo).

Possono essere di due tipi: reale in doppia precisione o alfanumerica (stringa di caratteri).

Una variabile alfanumerica è seguita dal simbolo $ e può contenere fino a 7 caratteri, eccetto la variabile "esclusiva" $ che può contenerne fino a 30.

TIPO	ESEMPIO
reale in doppia precisione numeri compresi tra ± 1 * 10 ↑ —99 e ± 9.999999999 * 10 ↑ 99	A=987.36
alfanumerico	A$="CASIO"

VETTORI

Un vettore si può definire come un insieme ordinato di elementi ognuno dei quali accessibile singolarmente. La variabile che individua l'elemento viene

indicata da una lettera seguita da un numero (racchiuso fra parentesi) che ne indica il posto all'interno del vettore stesso

A(3)   B(5)   Z(12)   ecc.

Queste variabili, come quelle usuali costituite dalla sola lettera, possono contenere sia valori numerici che, fatte seguire dal segno $, valori alfanumerici.

Così, volendo assegnare la stringa "VENERDI" al quinto elemento del vettore A scriverò

A$(5)="VENERDI"

Per la particolare gestione della memoria i nomi delle variabili corrispondono ad indirizzi fisici ben precisi e se ad esempio si è nella ripartizione DEFM 0, le variabili sono 26 e tali restano in qualsiasi modo siano chiamate.

Così ad esempio C sarà la stessa cosa di B(1) e di A(2).

Occorre fare molta attenzione a questa caratteristica, fonte potenziale di svariati errori.

Ecco di seguito la tabella delle "collisioni" che si verificano

```
A=A(0)
B=A(1)=B(0)
C=A(2)=B(1)=C(0)
.
.
.
.
Z=A(25)=B(24)=C(23)=.......=Y(1)=Z(0)
```

ISTRUZIONI DI INGRESSO DATI

INPUT "messaggio", var 1, var 2,....

Questa istruzione permette di inserire uno o più dati durante l'esecuzione di un programma. I nomi delle variabili a cui devono essere assegnati i dati vanno separati dalla virgola (,). Il messaggio può essere omesso o può essere ripetuto per ogni variabile

```
10 INPUT "NOME",A$, "COGNOME",B$
```

Quando si incontra uno statement di INPUT l'elaborazione si ferma e appare un "?" indicante l'attesa dell'immissione dei dati. Per riprendere l'esecuzione del programma si dovrà premere EXE.

In caso di immissione di più dati, questi non saranno inseriti uno di seguito all'altro, ma soltanto al momento opportuno, dopo la comparsa del rispettivo (?)

Essendo in attesa dell'inserimento di un dato, per uscire dal programma non è sufficiente premere AC ma si deve impostare MODE 0. Oltre a costanti numeriche o alfanumeriche è possibile inserire anche variabili ed espressioni; ad esempio si potrà avere

```
10 A=3
20 INPUT B
30 PRINT "B=";B
RUN
? A+5 (EXE)
B=8
```

FUNZIONE KEY

Viene usata tutte le volte che si desidera una determinata azione del computer in seguito alla pressione di un tasto.

Differisce da INPUT poichè il programma non si ferma nella condizione di attesa, ma prosegue nella sua elaborazione.

In pratica incontrando la parola chiave KEY il programma "legge" il carattere premuto in quell'istante sulla tastiera ed agirà di conseguenza, proseguendo anche nel caso in cui non venga premuto nulla.

Per questo motivo spesso si utilizza una linea del tipo

```
10 IF KEY="" THEN 10
```

che causa un loop fino a che non venga premuto qualcosa.

L'arresto che si determina differisce però in modo sostanziale da quello dell'istruzione INPUT, non comparendo il segno (?)

È bene rimarcare che KEY fornisce una lettura "istantanea" del carattere e un ulteriore utilizzo di quest'ultimo non sarà possibile se non provvedendo alla sua memorizzazione in una variabile stringa

```
10 A$ = KEY : IF A$ = " " THEN 10
20 V = VAL (A$)
30 PRINT V
```

IF...THEN... / IF... ; ...

L'istruzione permette, al verificarsi di certe condizioni, di eseguire alcune operazioni o di saltare ad un'altra linea del programma. Questi modi di utilizzo risultano ben distinti essendo presente in un caso il segno di (;) e nell'altro la parola chiave THEN. I due formati sono

IF (espressione di confronto) THEN (n° di linea o zona programma)
oppure
IF (espressione di confronto) ; (serie di istruzioni)

In entrambi i casi viene eseguito un test sulla espressione di confronto, detta anche CONDIZIONE. Se questa è vera si esegue ciò che è scritto dopo il (;) o viene effettuato il salto indicato dal THEN, mentre se è falsa si passerà alla riga successiva

```
10 INPUT A
20 IF A<0;PRINT "A<0":END
30 PRINT "A≥0"
```

```
10 INPUT A
20 IF A <0 THEN 40
30 PRINT"A≥0":END
40 PRINT "A<0"
```

Dopo il THEN si può introdurre un numero di linea oppure rimandare il programma ad un'altra zona (dalla 0 alla 9); in questo caso si dovrà far precedere la cifra scelta dal simbolo #.

GOTO...

Istruzione di salto incondizionato. Consente un rimando del programma alla linea o alla zona di programma indicata (da 0 a 9). Formato:

```
GOTO        (numero di linea)
GOTO        (espressione numerica)
GOTO        (# numero della zona)
GOTO        (# espressione numerica)
```

```
GOTO 100        salta alla linea 100
GOTO 2*50       salta alla linea 100
GOTO # 8        salta alla zona di programma 8
Se A=10
GOTO 2*A        salta alla linea 20
GOTO # A-3      salta alla zona di programma 7
```

CONCETTO DI CONTATORE

Dovendo sommare in un programma (Modo WRT) cinque numeri immessi dalla tastiera un primo tentativo potrebbe essere

```
10 INPUT A
20 INPUT B
30 INPUT C
40 INPUT D
50 INPUT E
60 S=A+B+C+D+E
70 PRINT S
```

ovviamente ciò è molto scomodo ed è più conveniente utilizzare un "contatore" che indichi il numero delle volte che deve essere ripetuta una serie di istruzioni. L'uscita dal "ciclo" avverrà in seguito ad un test sul valore raggiunto dal contatore; si terminerà quando questi ha assunto il valore 5

```
10 I=0: S=0        (inizializzazione del contatore I)
20 I=I+1: INPUT A      (incremento del contatore)
30 S=S+A
40 IF I<5 THEN 20      (test su I)
50 PRINT S
```

In questo modo però i valori immessi non vengono mantenuti in memoria (occupano tutti la variabile A) e non sarebbe quindi possibile sfruttarli per altri calcoli. Si può ovviare all'inconveniente memorizzandoli in un vettore

```
10 I=0: S=0
20 I=I+1: INPUT A(I)
30 S=S+A(I)
40 IF I<5 THEN 20
50 PRINT S
```

Al termine della routine i numeri impostati sono stati salvati nelle variabili B C D E F (corrispondenti ad A(1)...A(5))

FOR...NEXT

Benchè con le istruzioni di assegnazione e di salto sarebbe possibile programmare tutti gli algoritmi immaginabili, questo statement permette di semplificare il listato in innumerevoli situazioni poichè spesso il computer deve eseguire un lavoro ripetitivo o iterativo.

L'istruzione, il cui formato è

```
FOR (contatore) = (val. iniz.) TO (val. fin.) STEP (incremento)
.
.
.
.
.
NEXT (contatore)
```

consente di eseguire una serie di operazioni per un determinato numero di volte.

Incontrando un FOR, la variabile contatore assume il "valore iniziale", quindi vengono eseguite le istruzioni comprese tra il FOR e il NEXT. Al termine, incontrando il NEXT, la variabile contatore viene incrementata di una quantità pari a quella specificata dopo la parola chiave STEP. Se il valore ottenuto non supera il "valore finale" vengono eseguite nuovamente le istruzioni comprese tra FOR e NEXT, mentre in caso contrario il programma proseguirà con la prima istruzione successiva al NEXT.

```
10  FOR I=1 TO 10 STEP 2
20  PRINT I;
30  NEXT I
RUN
1 3 5 7 9
```

Nel caso di un incremento uguale ad 1 l'opzione STEP può essere omessa

```
10  FOR I=0.5 TO 10
20  PRINT I;
30  NEXT I
RUN
0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5
```

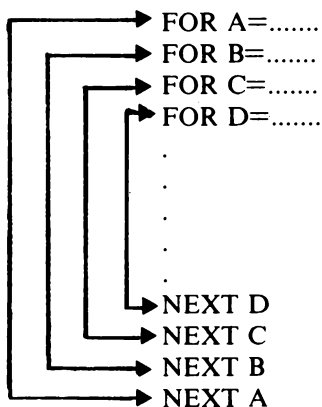
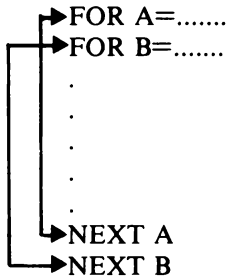
Sono possibili anche decrementi (incrementi negativi) ed in tal caso il ciclo

terminerà quando il contatore raggiunge un valore minore a quello di "valore finale"

```
10  FOR I= 7 TO 3 STEP -1
20  PRINT I;
30  NEXT I
RUN
7 6 5 4 3
```

Utilizzando la possibilità del ciclo, l'esempio fatto nel paragrafo precedente, sulla somma di cinque numeri, diventa:

```
10  S=0: FOR I=1 TO 5
20  INPUT A(I)
30  S=S+A(I)
40  NEXT I
50  PRINT S
```

Si possono usare fino ad un massimo di quattro cicli concatenati fra loro (nesting)

Non sono possibili accavallamenti fra i vari cicli

PRINT

Questa istruzione viene usata per far apparire sul display tutto ciò che si desidera.

PRINT 5	stampa il numero 5
PRINT A	stampa il valore di A
PRINT C*2+3	stampa il risultato dell'espressione
PRINT B$	stampa il contenuto della variabile B$

Quando, dopo l'istruzione PRINT, il calcolatore trova il carattere (") sa che deve stampare fedelmente tutto ciò che si trova racchiuso fra gli apici

PRINT "AREA DEL QUADRATO" (stampa la scritta)

È possibile inserire un massimo di 30 caratteri; una scritta più lunga farà comparire il messaggio di errore 2.

La visualizzazione di più valori o scritte si ottiene semplicemente mediante i separatori (,) o (;)

Usando il (;) i dati verranno stampati uno di seguito all'altro, mentre la (,) determinerà una sorta di riposizionamento a capo del cursore. La stampa verrà effettuata in quest'ultimo caso soltanto premendo EXE e sarà preceduta dalla completa cancellazione della scritta precedente

```
10  PRINT "MOLTIPLICO";A;" PER";B;
```

Ponendo ad esempio A=5 e B=6 il RUN fornirà

```
MOLTIPLICO 5 PER 6
```

```
10  PRINT "MOLTIPLICO",A," PER",B
RUN
MOLTIPLICO   (EXE)
5            (EXE)
PER          (EXE)
6
```

Desiderando ottenere la stampa in una precisa posizione nel pannello, si farà seguire alla parola PRINT la funzione CSR.

Come ben saprai il piccolo schermo del CASIO risulta diviso in 12 "spazi", numerati da 0 ad 11 partendo da sinistra.

Volendo ad esempio stampare la scritta "FIORE" al centro del display si opererà in questo modo

10 PRINT CSR 4; "FIORE"   e sul pannello apparirà

La funzione CSR può anche essere seguita da una variabile numerica o da un'espressione

```
PRINT CSR D;
PRINT CSR 10/2+3;
```

Bisogna tenere presente che la stampa di un valore numerico richiede sempre un ulteriore spazio riservato al segno, così con

PRINT CSR 4; 9

il numero 9 verrà visualizzato nella sesta posizione di stampa

Ultima osservazione riguarda la cancellazione e la "pulizia" del pannello.

Questa si ottiene utilizzando la semplice istruzione PRINT senza alcun dato o messaggio da stampare (PRINT).

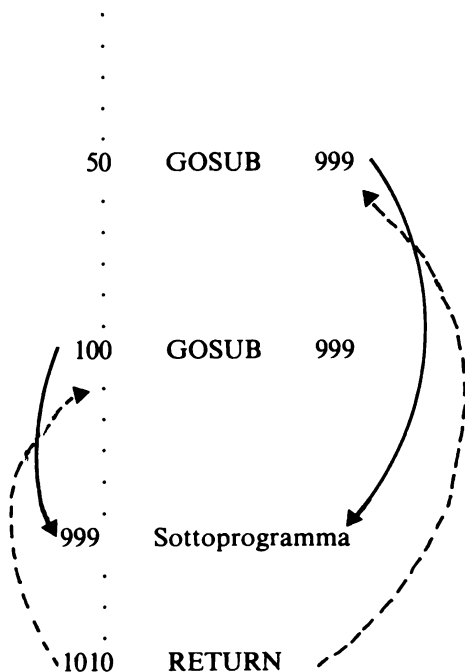
GOSUB....RETURN

Questo statement permette la stesura di un programma meglio articolato ed il risparmio di alcuni passi di memoria.

Dovendo inserire più volte la stessa sequenza di istruzioni è vantaggioso, invece di batterla ripetutamente, scriverla sotto forma di subroutine, che verrà richiamata dal programma principale ogni qual volta si renda necessario.

Questo sottoprogramma può essere posto in qualsiasi parte del listato, secondo la numerazione che ne risulta, o anche in un'altra zona di programma, con numeri di linea indipendenti.

L'accesso alle istruzioni in essa contenute avviene tramite il comando GOSUB seguito dal numero di linea in cui inizia la subroutine, o dall'indicazione della zona di memoria (GOSUB # 7).

Ciò produce un salto simile a quello del GOTO, l'esecuzione del sottoprogramma e il ritorno alla routine principale non appena si incontra lo statement RETURN; questo nuovo salto riporterà l'elaborazione all'istruzione immediatamente successiva al GOSUB di chiamata.

Il numero di linea che segue la parola chiave GOSUB può essere indicato anche con variabili o espressioni numeriche.

Ogni sottoprogramma può a sua volta richiamarne un altro fino ad esaurire il numero massimo delle concatenazioni permesse, qui pari a otto.

ELENCO ALFABETICO DI ALTRI COMANDI E FUNZIONI

CLEAR	cancella la zona di programma corrente. Deve essere eseguito solo in MODE 1
CLEAR A	cancella tutte le zone di programma
END	causa la fine dell'esecuzione del programma
LEN	fornisce la lunghezza di una variabile stringa [LEN(A$)]
LIST	permette la visualizzazione del listato, eventualmente a partire dalla linea indicata [LIST ... LIST 10]
LIST A	mostra il listato di tutte le zone di programma
MID	MID(n,p) estrae dalla variabile $ p caratteri a partire dall'n-esimo MID(n) estrae tutti i caratteri di $ a partire da quello numerato con n
MODE	seleziona uno o più dei 10 modi di funzionamento del computer
RAN #	genera un numero casuale compreso tra 0 ed 1
RND	RND(x,y) arrotonda x alla posizione numero y (-100<y<100)
RUN	dà il via all'esecuzione del programma, eventualmente a partire da una ben determinata linea
SET	stabilisce il numero delle cifre decimali (SET Fn) o di quelle effettive (SET En) di un valore numerico prima di mostrarlo sul pannello n deve essere compreso fra 0 e 9 SET N permette il ritorno alle condizioni iniziali
STOP	determina l'arresto dell'elaborazione che verrà ripresa premendo il tasto EXE
VAL	converte una variabile stringa formata da cifre nel corrispondente valore numerico
VAC	azzera il contenuto delle variabili numeriche e annulla quelle di carattere

FUNZIONI MATEMATICHE CHE COMPAIONO NEI PROGRAMMI

Funzione	Descrizione
ABS(X)	fornisce il valore assoluto di X ABS(28) ------28 ABS(-28)------28
COS(X)	coseno di X COS(90)------0
FRAC(X)	annulla la parte intera di un numero FRAC(123.456)------0.456
INT(X)	fornisce la parte intera di un numero INT(123.456)------123
SGN(X)	restituisce uno dei valori +1, -1, 0 a seconda del segno di X X>0------+1 X=0------0 X<0------ -1
SIN(X)	seno di X SIN(90)------ 1
SQR(X)	radice quadrata di X (X⩾0) SQR(81)----- 9

APPENDICE B

TRUCCHI E PARTICOLARITA'

Il Basic del PB-100 è molto completo ma, specialmente in un pocket, non si può certo pretendere la perfezione. Ogni computer ha delle caratteristiche che mancano negli altri e, spesso, delle lacune non presenti altrove.

Una delle istruzioni che mancano al CASIO è la REM (remark) che, per chi non la conoscesse, serve ad introdurre commenti nel listato, commenti che non influiranno sul corretto (si spera) funzionamento del programma. Questa documentazione può risultare molto utile sia a chi si ritrova fra le mani un listato fatto da altri, sia all'autore stesso allorchè, dopo un certo periodo di tempo, intenda modificare un programma o soltanto ricordare quale funzione svolge una certa subroutine.

Se ad esempio ci capita fra le mani il seguente programmino

```
10  INPUT A,B
20  IF A<B; C=A: A=B: B=C
30  Q=INT(A/B): R=A-B*Q
40  IF R≠0; A=B: B=R: GOTO 30
50  PRINT B
```

sarà ben difficile, ad un primo approccio, riuscire a comprendere che si tratta del calcolo del M.C.D. fra due numeri (con il noto Algoritmo di Euclide).

Se all'inizio del programma avessimo trovato una REM di spiegazione

```
5  REM M.C.D. FRA DUE NUMERI
10  INPUT A,B
..........................
..........................
```

il nostro compito sarebbe stato certamente facilitato.

La mancanza di una simile istruzione sul tuo pocket è certamente di poco conto senza l'espansione di memoria, sia perchè i programmi non potranno essere complicatissimi, sia perchè i commenti occupano un buon numero di passi che spesso risultano molto preziosi. Con l'aggiunta dell'espansione potrebbe viceversa essere senz'altro utile il poterla usare, ed a questo si può provvedere sfruttando una particolarità del computer.

Quando questi "legge" una istruzione di salto incondizionato (GOTO...) trasferisce l'esecuzione del programma alla linea indicata, senza preoccuparsi di ciò che segue. Facendo sempre riferimento al solito esempio potremmo scrivere

```
 5  GOTO 10: M.C.D. FRA DUE NUMERI
10  INPUT A,B
..................................................
..................................................
```

ed il programma, incredibile ma vero, funzionerà perfettamente. Provare per credere!

Ovvia raccomandazione è quella di ricordarsi dei due punti (:) dopo il numero di linea che segue il GOTO, altrimenti niente ti salverà dal logico e inevitabile ERR 2.

Spesso nella realizzazione dei giochi che ti abbiamo presentato ci siamo trovati in difficoltà per il semplice motivo che i 544 passi di memoria di cui il CASIO è fornito sembravano proprio insufficienti per riuscire a contenere tutto il programma elaborato. Ecco dunque che un risparmio anche di pochi passi risultava una vera necessità, e abbiamo dovuto sfruttare tutti gli accorgimenti possibili atti ad una riduzione più o meno consistente.

Innanzitutto molte volte abbiamo riunito in una sola riga quello che precedentemente era scritto in due o più di esse, ottenendo così il risparmio dei due passi di programma occupati dal numero di linea. Nel caso in cui le istruzioni contenute in una riga superassero le capacità della stessa (62 caratteri), abbiamo cercato di ovviare, per quanto possibile, eliminando tutti gli spazi e compattando la scrittura. Ciò si ottiene premendo il tasto DEL con il cursore posto direttamente sopra lo spazio da togliere.

Un altro accorgimento spesso usato è stato quello di dare una nuova numerazione al listato, inserendo dei numeri di linea più bassi (1,2,3,...); è infatti vero che per la loro scrittura sono necessari sempre 2 passi, come nella

numerazione 10,20,30,... ma si realizza un certo risparmio ad ogni istruzione di salto del tipo GOTO, GOSUB, THEN.

Per scrivere IF A=0 THEN 20 sono necessari 7 passi; IF A=0 THEN 2 ne occupa invece 6.

Anche alcuni accorgimenti nell'uso delle parentesi ti saranno utili per risparmiare qualche passo di programma.

Innanzitutto nota che esse si possono omettere in alcune funzioni dove normalmente vengono invece utilizzate, come SQR,INT,ABS,COS,SIN, SGN,ATN,EXP,LOG,TAN. Tieni però presente che tali omissioni non interferiscono nel risultato soltanto se il parametro utilizzato è un numero o una variabile numerica.

Nel caso in cui consista in un'espressione le parentesi saranno necessarie, altrimenti la funzione verrà calcolata soltanto sul primo dei valori che la seguono

```
SQR(84-3)       risultato 9
SQR 84-3        risultato 6.1651....
```

Sempre per quanto riguarda le parentesi c'è poi da notare che puoi tralasciare di chiuderle al termine dei vari calcoli senza che per questo compaia alcun messaggio di errore.
L'espressione

```
6+(4*3-(10-5*(4/2)))
```

darà lo stesso risultato di 18 anche se scritta nella forma

```
6+(4*3-(10-5*(4/2
```

e si risparmiano in questo modo tre passi. Nota che tutte le parentesi omesse vengono considerate chiuse al termine dell'espressione.

Un altro accorgimento al riguardo sarà possibile ricordando le notevoli capacità matematiche del tuo pocket.

Potrai ad esempio scrivere FOR I=1 TO 1 E 4 piuttosto che FOR I=1 TO 10000; due passi in meno sono sempre meglio di niente.

Procediamo ora verso considerazioni un po' più complesse.

Avrai certo notato che al computer sono spesso richiesti calcoli, operazioni e istruzioni che, più o meno simili si ripeteranno poi nel corso del programma. Comprenderai facilmente il vantaggio apportato dallo scrivere un'unica volta nel listato queste stesse operazioni, facendole tuttavia rieseguire ogni qual volta esse vengano richiamate. Parliamo, avrai capito, della subroutine,

a cui si accede tramite l'istruzione GOSUB e che termina con quella di RETURN.

Vediamone ora un uso pratico, nel caso più complesso in cui le operazioni che il computer deve eseguire più volte non siano perfettamente uguali.

```
1  PRINT "COMPARE IL NUMERO 1";
2  FOR I=1 TO 200: NEXT I
3  PRINT
4  PRINT "COMPARE IL NUMERO 2";
5  FOR I=1 TO 200: NEXT I
6  PRINT
7  PRINT "COMPARE IL NUMERO 3"
```

Noterai anche ad una prima lettura che le linee 2,3 e 5,6 sono identiche e che in quelle 1,4,7 varia soltanto il numero da stampare, mentre il resto della scritta risulta uguale.

È possibile quindi riunire tutte queste istruzioni ripetitive in una subroutine. L'unico accorgimento sarà quello di sostituire ai valori costanti 1,2,3 una variabile numerica (in questo caso A).

```
1  A=1: GOSUB 4
2  A=2: GOSUB 4
3  A=3: GOSUB 4: END
4  PRINT "COMPARE IL NUMERO"; A;
5  FOR I=1 TO 200: NEXT I
6  PRINT: RETURN
```

Un rapido calcolo ti permette di verificare che la prima stesura comporta un impiego di 113 passi di programma, mentre nella seconda versione ne sono stati utilizzati soltanto 75.

Se nonostante questi accorgimenti lo spazio non fosse ancora sufficiente alla memorizzazione del tuo programma, tieni infine presente la possibilità, da noi spesso sfruttata, di caricare le costanti necessarie in Modo RUN. Ciò può essere fatto prima o dopo la battitura del programma, ma naturalmente deve sempre precedere la sua esecuzione e consente un notevole risparmio di memoria anche se spesso, a causa della variazione dei valori inizialmente impostati, si è costretti a ripetere la suddetta procedura ad ogni nuovo start.

Le istruzioni FOR....NEXT si usano nei casi in cui si conosce il numero delle iterazioni da eseguire. Se non si conosce tale valore ma è noto il criterio

di arresto, in altri linguaggi di programmazione e talvolta in altri Basic vi sono parole chiave apposite che risolvono il problema.

Comunque è possibile ovviare a tale mancanza facendo uso anche in questi casi dell'istruzione FOR.

Due sono le possibili soluzioni:

a) prevedere un numero molto elevato di iterazioni, superiore a quello necessario, valutando la condizione di arresto internamente al ciclo
b) utilizzando un ciclo con passo (STEP) nullo

Esempio: supponiamo di dover ricercare il minimo fra una serie di numeri impostati sulla tastiera. Il termine dei valori verrà segnalato battendo il numero 0, e naturalmente non si conosce precedentemente quanti saranno i numeri che dovremo battere.

Secondo quanto detto, due routine potrebbero essere

a)
```
10  M=1000
20  FOR I=1 TO 100000
30  INPUT N
40  IF N=0 THEN 70
50  IF N<M; M=N
60  NEXT I
70.......
```

b)
```
10  M=1000
20  FOR I=1 TO 2 STEP 0
30  INPUT N
40  IF N=0 THEN 70
50  IF N<M; M=N
60  NEXT I
70  ...........
```

Sinceramente noi, fra le due alternative, preferiamo la seconda.

La funzione VAL ti permette, come sai, di convertire una variabile stringa nel suo corrispondente valore numerico.

Esempio: 100 Y=VAL(A$)
Se A$ = "345" si ottiene Y=345

Potrebbe talvolta essere utile il poter disporre della funzione inversa, che dovrebbe convertire il valore numerico in una stringa. Questo purtroppo non è possibile, ma avendo a che fare con un numero di una sola cifra il problema è presto risolto.

Supponiamo di dover assegnare ad A$ la stringa corrispondente alla cifra (da 0 a 9) scelta casualmente tramite la funzione RAN#.

Si opererà in questo modo

```
10  $="0123456789"
20  C=INT(RAN# * 10)
30  A$=MID(C+1,1)
```

e il gioco è fatto.

Un altro esempio che si basa sulla solita tecnica è il seguente, che permette di proseguire soltanto immettendo, tramite la funzione KEY, un valore numerico

```
10  $= "0123456789"
20  IF KEY="" THEN 20
30  FOR I=1 TO 10
40  IF KEY≠MID(I,1); NEXT I: GOTO 20
50  ................
```

Un'operazione in realtà possibile, come l'elevamento a potenza di un numero negativo, non può su questo pocket essere eseguita liberamente poichè fornisce un risultato spesso inattendibile.

Ad esempio -3 ↑ 2 ha come effetto -9

Si può ovviare all'inconveniente eseguendo il calcolo tramite variabili A=-3 A ↑ 2 ------- +9
o, ancor meglio, facendo uso di parentesi

(-3)↑2 ------ +9

Fra le istruzioni meno comuni del tuo pocket ha un certo rilievo la funzione di arrotondamento RND. Talvolta però non si desidera che un valore venga arrotondato, ma si vuol ottenere il suo troncamento ad una determinata cifra decimale.

Questa operazione non è ottenibile con una singola istruzione e se ad esempio si dovrà troncare il valore della variabile A alla seconda cifra decimale dovremo fare

INT (A * 1 E 2) / 1 E 2

In questo modo il valore di A verrà dapprima moltiplicato per 100, quindi resterà formato dalla sola parte intera ed infine, diviso per 100, fornirà il risultato voluto.

Per concludere questa appendice ti proponiamo un semplice programma che ti permetterà di conoscere tutte le cifre decimali di un quoziente, in "barba" alla massima capacità del computer che prevede per il calcolo interno una mantissa di 12 posizioni

```
10  INPUT "DIVIDENDO",X
20  INPUT "DIVISORE",Y
30  Q=INT(X/Y)
40  X=(X-Y*Q)*10
50  PRINT Q;: GOTO 30
```

APPENDICE C

MESSAGGI D'ERRORE

Uno degli aspetti più frequenti e senz'altro meno simpatici della programmazione è il riconoscimento degli errori con le relative correzioni. Certamente innumerevoli volte sarai incorso in distrazioni o sbagli che hanno determinato la comparsa sul pannello dell'odiato messaggio e poichè questo argomento ci sembra spiegato un po' troppo succintamente nel tuo manuale, riteniamo opportuno ed utile cercare di approfondire alcune situazioni di errore.

Un primo consiglio che possiamo darti è quello di controllare in modo RUN il contenuto delle variabili sospette nel caso in cui il messaggio non ti dia le informazioni sufficienti per la correzione.

Il CASIO ti offre inoltre la possibilità, come risulta chiaro dal manuale, di usare opportunamente l'istruzione di STOP o il modo TRACCIA (MODE 2). Tieni però presente che quest'ultimo non può essere usato liberamente per il debugging di programmi che contengono la funzione KEY. Ad esempio avendo nel listato

```
10 IF KEY="" THEN 10
```

premendo RUN (EXE) si avrebbe una attesa "infinita" della pressione di un tasto e l'esecuzione si arresterebbe alla linea in questione.

ERRORE 1

Questo messaggio compare quando nello scrivere un programma si supera la capacità di memoria disponibile.

Ciò può accadere o perchè si eccedono i passi consentiti (544 con 26 variabili) oppure quando si imposta un numero di variabili troppo alto rispetto ai byte necessari a tale operazione. È evidente che avendo ad esempio a disposizione 63 passi di programma, poichè occorrono 8 byte per ogni

variabile aggiunta, è possibile incrementare le memorie al massimo di 7 unità.

Per ovviare all'errore 1, se ritieni che l'eccedenza dei passi del tuo programma sia abbastanza limitata, puoi cercare di occupare meno memoria seguendo anche gli accorgimenti descritti nell'Appendice B. Se l'errore si riferisce alla elevata richiesta di variabili un altro consiglio, seppur banale, è quello di utilizzare più volte la stessa variabile non appena essa si rende disponibile.

Il messaggio d'errore qui descritto si verifica anche quando si superano, nella stesura di formule, i sei livelli di parentesi consentiti

A=1+(2+(3+(4+(5+(6+(7+(8+9)))))))

In questo caso infatti il computer è costretto a tenere in sospeso i calcoli oltre la sua capacità di accumulo e non rimane altro che semplificare la scrittura dell'espressione numerica.

ERRORE 2

Probabilmente si tratta dell'errore più frequente in cui puoi incorrere. Molto spesso è infatti causato da semplici distrazioni nella stesura del programma, come potrebbe essere l'errata ortografia di una parola chiave, la dimenticanza del segno di (:) tra una istruzione e l'altra, l'omissione del segno $ nelle variabili stringa.

Un altro errore di questo tipo è quello di scrivere GOTO dopo il THEN in una istruzione condizionale

IF A = 5 THEN GOTO 30

o ancora fare una cosa del tipo

IF A = 5; 30

Sul pannello comparirà sempre questo errore di sintassi se tenterai di scrivere in una istruzione di stampa un messaggio superiore ai trenta caratteri. In tal caso occorre dividere la scritta in due o più parti tenendo presente che, se occupa una sola riga, non è necessario ripetere l'istruzione PRINT.

Stranamente si incorre in questo errore anche quando si applica la funzione VAL ad una variabile stringa considerata come elemento di un vettore:

C=VAL(A$(3))   oppure   C=VAL(A$(I))

Nel primo esempio si può ovviare sostituendo ad A$(3) il corrispondente D$; nel secondo è necessaria la seguente variazione:

```
B$=A$(I):C=VAL(B$)
```

Una particolarità è che in questi due casi non appare il numero di linea nel quale si verifica l'errore.

Più difficili da correggere sono errori concettuali che riguardano il diverso tipo di variabile (numerica o alfanumerica) a sinistra e a destra in uno statement di assegnazione.

Si può incorrere in un simile sbaglio tentando ad esempio di ottenere l'inverso della funzione VAL che, ricordiamo, non è disponibile sul CASIO e per la quale rimandiamo all'appendice B.

Cercando di assegnare ad A$ la stringa corrispondente al valore numerico contenuto in B

```
A$=B
```

si verificherà appunto tale errore.

ERRORE 3

L'errore 3 comparirà principalmente cercando di effettuare una operazione matematicamente impossibile, oppure usando un argomento non accettabile per le varie funzioni matematiche perchè fuori dalla loro gamma input.

Classici esempi sono nel 1° caso la divisione per 0 e nel secondo la radice quadrata di un numero negativo.

L'elevamento a potenza è qui possibile soltanto con basi positive; in caso contrario, pur non comparendo il messaggio di errore, il risultato non è attendibile.

L'errore matematico si avrà anche se durante un calcolo si supera in valore assoluto il numero $9.999999999 * 10^{99}$ che corrisponde alla massima capacità del computer.

ERRORE 4

Si verifica quando dopo istruzioni di salto come GOTO, GOSUB, THEN si trova un numero che non corrisponde ad alcuna linea del programma

```
10   IF A=0 THEN 20
```

Se A=0 e non esiste alcuna linea 20 comparirà il messaggio di errore. Questa distrazione è più probabile in seguito ad una nuova versione del tuo programma con conseguente modifica della numerazione delle linee.

ERRORE 5

Questo messaggio viene visualizzato quando nell'esecuzione di alcuni comandi si usano argomenti che non rientrano tra quelli possibili.

Il tentativo di estrarre un numero di caratteri superiore a quelli contenuti in $ ti farà incorrere in questo errore.

Altro esempio può essere quello di usare la funzione di arrotondamento RND(x,y) con y⩾100 o y⩽-100

Un caso che a prima vista può sembrare banale è quello di cercare di ottenere la stampa, tramite l'istruzione CSR, in uno spazio non compreso fra 0 e 11.

Questo può accadere facendo uso di variabili che, durante i calcoli, assumono valore negativo. Se ad esempio A=-1 l'istruzione

PRINT CSR A; $;

causerà l'errore 5 mentre stranamente qualora il valore di A superi 11 la stessa istruzione darà ERRORE 2.

ERRORE 6

È un errore che riguarda principalmente l'uso scorretto delle variabili. Si verifica quando inavvertitamente si usa la stessa memoria per la variabile numerica e la variabile stringa che come saprai non possono coesistere con lo stesso nome.

Assegnando ad esempio il carattere "3" ad A$ risulterà errata una linea del tipo

IF A=3 THEN 100

Sarà quindi necessario il cambiamento del nome della variabile o la ridefinizione della stessa.

L'errore 6 si avrà anche nel tentativo di usare una variabile non disponibile. Ad esempio nella situazione di ripartizione di base (26 memorie e 544 passi) non sarà utilizzabile Z(1).

Un altro motivo di errore è rappresentato dall'accumulo in una variabile stringa di un numero troppo elevato di caratteri. In questo senso

```
10  A$=""
20  FOR I=0 TO 7
30  A$=A$+"1"
40  NEXT I
```

è una routine non corretta in quanto A$ può contenere soltanto 7 caratteri.

Una situazione che causa non soltanto l'errore in questione ma addirittura il blocco del sistema (ogni richiesta del contenuto delle variabili darà ERR 6) è quella riguardante il VAC.

Infatti questo comando, che annulla tutte le variabili, deve sempre essere scritto da solo in una linea senza essere seguito da nessun'altra istruzione.

È bene attenersi a questa avvertenza anche se non sempre la sua mancata applicazione causa l'errore.

In presenza del "blocco" non resta altro che premere il pulsante di ALL RESET posto sul retro con un oggetto appuntito.

Ciò comporta il ritorno alla normalità con tutta la memoria azzerata; si potrà quindi riscrivere il programma con un uso corretto del VAC.

ERRORE 7

Quante volte ti è capitato di dover controllare il programma in seguito a questo messaggio e poi, dopo l'ennesimo controllo, arrenderti sconsolato essendo in teoria tutto perfettamente regolare? Se non sei mai incorso in questa situazione sei veramente un'eccezione.

Tieni infatti presente che oltre agli effettivi errori riportati sul manuale:

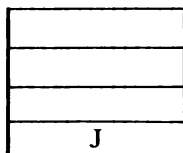
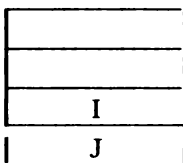
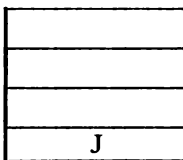
- — superamento dei quattro annidamenti consentiti per il FOR o degli otto previsti per la subroutine
- — NEXT che non fa riferimento ad alcun FOR
- — RETURN senza GOSUB
- — variabile del NEXT differente da quella usata per il FOR

tale messaggio può verificarsi anche a causa di un inconveniente molto fastidioso e che spesso passa inosservato.

Frequentemente capita di dover uscire da un ciclo FOR prima della sua fine logica, cioè prima che l'indice abbia raggiunto il limite massimo e questo

può portare all'errore in questione, ad esempio se si incontra un altro FOR che usa la stessa variabile come contatore. Vediamo di capirne il perchè: ogni volta che nel programma si incontra un ciclo FOR la variabile-contatore viene posta in una pila, ed ogni volta che un ciclo viene chiuso la rispettiva variabile viene tolta dalla pila. Il tutto per controllare i livelli di "annidamento" dei cicli che, ricordiamo, nel CASIO possono essere soltanto quattro. Vediamo con un esempio

```
10  FOR J=1 TO 5
20  FOR I=1 TO 10
30   IF A(I)=0 THEN 50
40  NEXT I
50  NEXT J
```

Quando si incontra il ciclo FOR nella linea 10 la variabile J viene posta nella pila

e poi, all'apertura del nuovo ciclo nella 20 anche I verrà posta in essa

Quando alla linea 40 termina il ciclo su I la rispettiva variabile viene tolta dalla pila che rimane formata dal solo J.

Tuttavia nel caso si esca dal ciclo prima della sua fine logica, cioè se uno degli A(I) è uguale a zero, il contatore stesso non verrà tolto creando dei problemi in caso di annidamenti in cui risulti utilizzata la stessa lettera.

Ecco una routine che causa l'errore

```
10  FOR A=1 TO 10
20  IF A=5 THEN 40
30  NEXT A
40  FOR B=1 TO 10
50  FOR A=1 TO 3
60  PRINT B↑A;
70  NEXT A
80  NEXT B
```

Due soluzioni per evitare inconvenienti di questo genere sono:

a) non usare il ciclo FOR che causa l'errore incrementando e controllando "manualmente" la variabile indice

Invece di

```
10 FOR A=1 TO 10
20 IF A=5 THEN 40
30 NEXT A
```

fare

```
5  A=0
10 A=A+1
20 IF A=5 THEN 40
30 IF A<10 THEN 10
```

b) "ingannare" il computer modificando, all'interno del ciclo, il valore dell'indice ponendolo uguale al limite superiore prima di uscire

Invece di

```
10 FOR A=1 TO 10
20 IF A=5 THEN 40
30 NEXT A
```

fare

```
10 FOR A=1 TO 10
20 IF A=5; A=10
30 NEXT A
```

Tutti i programmi sono stati controllati e verificati più volte. Ciò non toglie che qualcosa, oltre ad eventuali errori di stampa, possa essere sfuggito.

Per qualunque chiarimento in proposito potete scrivere o telefonare a:

Del Bello Sandro - Paganini Anna
Via Genova 114 - 19100 La Spezia
Tel. (0187) 701136

GRUPPO EDITORIALE JACKSON

Sandro Del Bello e Anna Paganini

30 GIOCHI PER IL CASIO

218